

Grado Universitario en Ingeniería en Tecnologías  
Industriales  
2018-2019

*Trabajo Fin de Grado*

# “Asistente conversacional para la mejora de la accesibilidad”

---

Alejandro Pérez Dueñas

Tutor/es

David Griol Barres

Universidad Carlos III de Madrid 2 de Julio de 2019



Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento - No Comercial - Sin Obra Derivada**



## RESUMEN

En este Trabajo de Fin de Grado se desarrolla un agente conversacional, integrado en una aplicación creada para ello, con el objetivo de proporcionar información a los ciudadanos acerca de la accesibilidad de estaciones y edificios en la Comunidad de Madrid. Este proyecto se enmarca dentro de una estrategia Europea que tiene como fin el conseguir ciudades inteligentes.

El concepto de *Smart City* ha sido extensivamente investigado, y en esta memoria se presentan sus principales aspectos, características y objetivos. No es un concepto único, ni se refiere tan solo a una disciplina o un aspecto de la sociedad o las ciudades, si no que engloba una gran variedad de medidas y políticas diferentes. Pese a esto, podríamos decir que las ciudades inteligentes son aquellas ciudades que mediante el uso de las tecnologías de la comunicación y de la información alcanzan un nivel mayor de sostenibilidad y proporcionan una mayor calidad de vida a sus habitantes.

Se indaga también en la versatilidad que los agentes conversacionales ofrecen, quedando en manos del desarrollador el pensar en posibles escenarios en los que puedan resultar útiles. Su implantación resulta especialmente beneficiosa cuando se trata de proporcionar información general o acerca de servicios a usuarios. Esto es debido a la posibilidad de conectarlos a bases de datos o servidores externos donde podrán realizar consultas a dicha información.

Finalmente, se hace un estudio económico del coste asociado a la creación de la solución propuesta en esta memoria, así como un estudio de las posibilidades económicas del proyecto a futuro y de su situación con respecto a la legislación vigente.

**Palabras clave:** asistente virtual; aplicación móvil; movilidad; ciudad inteligente; sostenibilidad.



## **DEDICATORIA**

En primer lugar dar las gracias a mis padres y a mi hermana por su ayuda y apoyo diario.

Agradecer también a la Universidad y a mi tutor David Griol por la ayuda y el tiempo dedicado en este proyecto.



## ÍNDICE GENERAL

1. INTRODUCCIÓN. . . . .	1
1.1. Motivación . . . . .	1
1.2. Objetivo . . . . .	2
1.3. Estructura de la memoria . . . . .	3
2. ESTADO DEL ARTE. . . . .	4
2.1. Sistemas de dialogo hablado . . . . .	4
2.1.1. Precedentes de los asistentes conversacionales. . . . .	5
2.1.2. Situación económica. . . . .	6
2.1.3. Situación actual. . . . .	7
2.1.4. Ejemplos de asistentes conversacionales en el sector público. . . . .	9
2.1.5. Servicios web para el desarrollo de asistentes. Dialogflow. . . . .	9
2.2. Concepto de Smart Cities. . . . .	12
2.2.1. Características de las ciudades inteligentes. . . . .	13
2.2.2. Smart Cities y Europa 2020. . . . .	15
2.2.3. Ciudades inteligentes y ciudades accesibles. . . . .	16
2.2.4. Proyecto SIMON. . . . .	17
2.3. Sistemas operativos móviles. . . . .	18
2.3.1. Principales sistemas operativos. . . . .	18
2.3.2. Arquitectura de Android. . . . .	20
2.3.3. Arquitectura iOS. . . . .	23
3. DISEÑO Y DESARROLLO DEL SISTEMA. . . . .	25
3.1. Descripción de las funcionalidades básicas. . . . .	25
3.2. Agente conversacional. . . . .	27
3.2.1. Conversación entre agente y usuario. . . . .	28
3.2.2. Intenciones o <i>intents</i> . . . . .	30
3.2.3. Entidades o <i>Entities</i> . . . . .	34
3.2.4. Integración. . . . .	35

3.3. API REST. . . . .	35
3.3.1. Datos Abiertos del Ayuntamiento de Madrid. . . . .	36
3.4. Aplicación. . . . .	37
3.4.1. Herramientas utilizadas. . . . .	37
3.4.2. Presentación de la aplicación. . . . .	43
3.4.3. Pantalla de conversación. . . . .	45
3.4.4. Pantalla de lugares favoritos. . . . .	47
3.4.5. Pantalla de mapas. . . . .	49
3.5. Evaluación. . . . .	49
3.5.1. Análisis de los resultados . . . . .	56
3.5.2. Análisis de la UAT . . . . .	63
4. GESTIÓN DEL PROYECTO . . . . .	65
4.1. Planificación temporal . . . . .	65
4.2. Marco Regulador. . . . .	67
4.3. Presupuesto . . . . .	68
4.3.1. Costes directos . . . . .	69
4.3.2. Costes indirectos . . . . .	70
4.3.3. Costes totales . . . . .	70
4.4. Entorno socio-económico . . . . .	71
5. CONCLUSIONES Y TRABAJO FUTURO . . . . .	73
5.1. Trabajo futuro . . . . .	74
BIBLIOGRAFÍA . . . . .	76





## ÍNDICE DE FIGURAS

2.1	Cuota de mercado de altavoces inteligentes . . . . .	7
2.2	Preguntas más frecuentes a los altavoces inteligentes . . . . .	8
2.3	Dialogflow: funcionamiento del fulfillment . . . . .	10
2.4	Funcionamiento de los intents . . . . .	11
2.5	Características de las Smart Cities . . . . .	14
2.6	Captura de pantalla aplicación SIMON . . . . .	17
2.7	Cuota de mercado Sistemas operativos . . . . .	19
2.8	Arquitectura android . . . . .	21
2.9	Arquitectura iOS . . . . .	23
3.1	Flujo de conversación entre el agente y el usuario . . . . .	29
3.2	Frases de entrenamiento . . . . .	31
3.3	Frases de entrenamiento con parámetros. . . . .	31
3.4	Respuestas programadas del agente . . . . .	32
3.5	Activación del <i>Fallback intent</i> . . . . .	33
3.6	Asistente de creación de entidades. Entidad distrito . . . . .	34
3.7	Bloque de lógica del asistente . . . . .	43
3.8	Flujo de la aplicación . . . . .	44
3.9	Pantalla de inicio . . . . .	45
3.10	Pantalla de conversación . . . . .	46
3.11	Pantalla de favoritos . . . . .	47
3.12	Menú desplegable al pulsar sobre favorito . . . . .	48
3.13	Pantalla de mapas . . . . .	49
4.1	Diagrama de Gantt . . . . .	66
4.2	Población con discapacidad en España . . . . .	72



## ÍNDICE DE TABLAS

2.1	Planes de uso de Dialogflow . . . . .	10
3.1	Tabla de pruebas. . . . .	50
3.2	P - 01: Iniciar aplicación desde iOS . . . . .	50
3.3	P - 02: Iniciar el asistente por texto . . . . .	51
3.4	P - 03: Iniciar el asistente por texto . . . . .	51
3.5	P - 03: Respuesta del asistente por texto . . . . .	51
3.6	P - 03: Recibir respuesta por voz . . . . .	52
3.7	P - 07: Consultar información sobre edificio . . . . .	52
3.8	P - 08: Consultar información sobre estación . . . . .	52
3.9	P - 09: Éxito en la consulta . . . . .	53
3.10	P - 10: Visualizar en el mapa . . . . .	53
3.11	P - 11: Añadir lugar a favoritos . . . . .	53
3.12	P - 12: Ir a la ventana de favoritos . . . . .	54
3.13	P - 13: Información sobre lugares favoritos disponible . . . . .	54
3.14	P - 14: Eliminar lugar de la lista de favoritos . . . . .	54
3.15	P - 15: Ir al mapa desde favoritos . . . . .	55
3.16	P - 16: Cancelar pulsación en pantalla de favoritos . . . . .	55
3.17	P - 17: Desplazarse por el mapa . . . . .	55
3.18	P - 18: Volver a la pantalla de conversación desde el mapa . . . . .	56
3.19	Resultado de las pruebas . . . . .	56
3.20	Terminales usados en UAT . . . . .	57
3.21	Pregunta 1 encuesta de evaluación . . . . .	58
3.22	Pregunta 2 encuesta de evaluación . . . . .	58
3.23	Pregunta 3 encuesta de evaluación . . . . .	59
3.24	Pregunta 4 encuesta de evaluación . . . . .	59
3.25	Pregunta 5 encuesta de evaluación . . . . .	60
3.26	Pregunta 6 encuesta de evaluación . . . . .	60

3.27	Pregunta 7 encuesta de evaluación . . . . .	61
3.28	Pregunta 8 encuesta de evaluación . . . . .	61
3.29	Pregunta 9 encuesta de evaluación . . . . .	62
3.30	Pregunta 10 encuesta de evaluación . . . . .	62
3.31	Pregunta 11 encuesta de evaluación . . . . .	63
4.1	Fechas de inicio y fin de las etapas del proyecto . . . . .	66
4.2	Estimación de horas empleadas . . . . .	67
4.3	Estimación del sueldo del empleado . . . . .	69
4.4	Estimación del coste del Hardware . . . . .	69
4.5	Estimación del precio del software . . . . .	70
4.6	Estimación de los costes totales . . . . .	70



# 1. INTRODUCCIÓN

En este primer capítulo de la memoria, se describe la motivación para realizar este proyecto, los objetivos planteados inicialmente y como han evolucionado con el desarrollo del trabajo, y la estructura que se seguirá en el documento.

## 1.1. Motivación

Según un informe de las Naciones Unidas, en torno al 55 % de la población en 2018 residía en áreas urbanas, y se espera que esta cifra aumente hasta el 68 % para el 2050. En el marco de la Unión Europea, estas cifras son aún mayores, y se sitúan en un 75 % actualmente, que escalaría hasta el 85 % en 2050 [1].

El éxodo de la población a las grandes urbes trae consigo una serie de problemas que deben ser paliados con medidas comunes. Un aumento de la población en las grandes ciudades implica un aumento de la concentración de la contaminación en las mismas, saturación de las carreteras y calles, complicaciones para acceder a los servicios básicos tales como pueden ser la sanidad o la educación, ya que sería necesario estructurar correctamente la ciudad para evitar estos problemas.

Al aumento de la población en las ciudades, hay que añadir el hecho de que la población está envejeciendo. Según el banco mundial, la esperanza de vida ha aumentado desde 1960 hasta la actualidad unos veinte años [2]. Los problemas relacionados con la movilidad aumentan con el aumento de la edad, por tanto, para que los ciudadanos más mayores puedan disfrutar de una calidad de vida es necesario que se produzcan cambios en la forma de vida de la sociedad y en cómo se construyen las ciudades.

Desde hace unos años hay un concepto que ha ido ganando peso en la sociedad, y es el concepto de ciudades inteligentes o *smart cities* [3]. Para lograr unas ciudades inteligentes son necesarias una serie de políticas y proyectos de diversos ámbitos, necesarias para encarar los problemas que el flujo de ciudadanos desde los pueblos a las ciudades puede acarrear.

El fin último de las ciudades inteligentes es el de mejorar la calidad de vida de los habitantes de la propia ciudad mediante el uso de las tecnologías de la comunicación y de la información. Se plantean medidas para que el crecimiento que se dé en los siguientes años sea lo más eficiente y sostenible posible. Consisten en medidas de planificación y organización de las ciudades para que los ciudadanos que habiten en ellas puedan disfrutar de la mayor calidad de vida. Las medidas han de ser transversales para poder conseguir el fin propuesto por las ciudades inteligentes. De nada serviría mejorar el transporte de una ciudad si esto viene acompañado de un detrimento en el cuidado del medio ambiente; o llevar a cabo medidas para paliar la contaminación si esto trae implícito problemas para

el correcto funcionamiento de la sociedad y las urbes en las que se encuentran.

## **1.2. Objetivo**

Este proyecto nace con el objetivo de mejorar la accesibilidad de estaciones y edificios en el marco de la Comunidad de Madrid. Nace como respuesta a un problema creciente de desconocimiento, por parte de la población y de los turistas que viajan hasta la Comunidad, acerca de la accesibilidad. Es tan importante disponer de los medios para facilitar la movilidad a los ciudadanos como que estos sean conscientes de la existencia de estas herramientas, su funcionamiento y el estado del funcionamiento en cada momento.

A menudo, se producen inconveniencias en el transporte público, y en los accesos al mismo. Para la gran mayoría de la población, unas escaleras mecánicas averiadas, o un ascensor no funcional, suponen un inconveniente solucionable. El verdadero problema viene cuando los usuarios sufren algún tipo de deficiencia en la movilidad, y necesitan de dichos elementos para poder acceder o salir de la estación o del edificio. El hecho de que el elemento que les permite desplazarse correctamente esté averiado trae unas consecuencias muy negativas para la calidad de vida, y el hecho además de que estos problemas sean inesperados y los propios ciudadanos no sean conscientes de los mismos crea a su vez un ambiente de molestar y de indignación que puede ser paliado mediante este proyecto.

Que la parte de la sociedad que más necesita elementos facilitadores de la movilidad tenga la posibilidad de acceder a ver el estado de dichos elementos de antemano, antes incluso de tomar la decisión de elegir un medio de transporte, o evitar un determinado día a un edificio municipal, puede ahorrar muchos problemas a estos usuarios.

Por tanto, los principales objetivos del proyecto son:

- Permitir el acceso a los usuarios a información acerca de la accesibilidad de estaciones y edificios en la Comunidad de Madrid.
- Permitir a los usuarios el conocer en tiempo real la situación en cada uno de los lugares de interés con información disponible.
- Recibir alertas en caso de que se produzcan incidencias.
- Almacenar sus lugares favoritos en una pantalla dedicada a ello para poder tener un acceso más rápido.
- Disponer de un mapa donde visualizar el lugar de la consulta.



### 1.3. Estructura de la memoria

Este documento está dividido en diferentes capítulos, con el objetivo de proporcionar al lector un análisis completo del proyecto, desde el contexto en el que se enmarca, hasta la explicación de ciertos detalles relevantes para el correcto desarrollo. Las secciones principales del documento son:

- **Introducción.** Esta misma sección tiene como objetivo el plantear las metas de este proyecto y la motivación para la creación del mismo.
- **Estado del arte.** Se realiza una explicación de los conceptos y tecnologías más relevantes e influyentes en este proyecto, así como un análisis comparativo de otros proyectos similares.
- **Diseño y desarrollo del sistema.** Se ofrece una explicación detallada del proceso de creación de todas las partes que componen este proyecto, desde el diseño de la aplicación hasta el proceso de creación del agente conversacional.
- **Gestión del proyecto.** Aquí se enumeran los recursos necesarios para realizar este proyecto, así como los costes asociados a cada uno de los recursos. Se incluye una explicación acerca del marco regulador y del entorno socio-económico.
- **Conclusiones y trabajo futuro.** Como parte final se ofrece un resumen de los conocimientos y del trabajo futuro que se podría llevar cabo para la ampliación y mejora del proyecto.

## 2. ESTADO DEL ARTE

Este capítulo tiene como objetivo el poner en un marco los principales temas que se tratan en este proyecto, su situación actual y una serie de ejemplos detallados. Podemos distinguir tres grandes bloques dentro del Estado del Arte. En primer lugar, se comenzará por desarrollar los sistemas de diálogo hablado, explicando qué son, sus objetivos y funcionalidades presentes y posibles expansiones en el futuro.

Más adelante, se enmarca el proyecto dentro de las *Smart Cities*. Se presenta a las ciudades inteligentes y se da una serie de ejemplos de proyectos similares que de igual manera se enmarcan dentro de este concepto.

Finalmente, ya que la aplicación estará disponible para más de un sistema operativo, se presentan los sistemas operativos más importantes y se indaga en los sistemas en los que se podrá usar la aplicación.

### 2.1. Sistemas de dialogo hablado

Un sistema de diálogo puede ser definido como una interfaz para la interacción entre humanos y máquinas. Esta interfaz tiene la posibilidad de reconocer y entender una introducción de texto u oral, a la cuál es capaz de responder con coherencia con el objetivo de proporcionar información o llevar a cabo una acción específica. La manera de cumplir el objetivo, será mediante la interacción en repetidas ocasiones en lenguaje natural con el usuario, mediante la que descubrirá dicho objetivo [4].

El objetivo de estos sistemas, es de simular un dialogo que el usuario podría tener con otro humano, con el fin de simplificar el modo en el que se produce la petición.

Para poder llevar a cabo esta conversación y poder asistirle, son necesarios una serie de módulos. El objetivo es recoger la información que el usuario proporciona, analizarla y responder de forma coherente al usuario, teniendo siempre en cuenta cuál es el objetivo de la interacción.

Los sistemas de dialogo hablado necesitan cinco módulos para poder procesar el lenguaje humano[5]. Los módulos son los siguientes:

- Sistema de reconocimiento automático del habla, también conocido como *Speech-to-text* (STT). El objetivo es recibir las consultas del usuario a través del lenguaje hablado por parte del usuario, y transformar esta consulta en un texto que podrá ser analizado más adelante por módulos posteriores.
- Comprensión del lenguaje hablado. Lo que se obtiene del módulo anterior, es lo que servirá de entrada para éste módulo. La comprensión del lenguaje tiene como objetivo realizar un análisis semántico del texto, para poder comprender qué ha

dicho el usuario. Para esto se analizan ciertas palabras clave que puede haber dicho el usuario en las sucesivas iteraciones de la conversación, para así poder poner en contexto la conversación.

- **Gestión del diálogo.** A partir del análisis realizado en el módulo anterior, en este módulo se decidirá cuál será la siguiente respuesta que el asistente le dé al usuario. Para poder responder al usuario con información realizará consultas en bases de datos y servidores externos, e incluirá dicha información en la respuesta final.
- **Generación del lenguaje natural.** El objetivo es transformar la información que será entregada al usuario a una o más frases que sean correctas gramatical y semánticamente, y que sean coherentes con el estado de la conversación actual. Para poder modelar la respuesta en lenguaje natural, este módulo realiza consultas a bases de datos donde tiene guardadas posibles respuestas, donde simplemente tendrá que introducir en la respuesta la información que ha recabado en módulos anteriores.
- **Habla de texto o *text-to-speech* (TTS).** Éste último módulo se encarga de leerle la respuesta creada en el módulo anterior al usuario, es decir, sintetizando el texto generado a voz. Es un módulo muy complejo debido a que en el lenguaje existen abreviaciones complicadas de leer o incluso números.

Como podemos comprobar, el flujo de la conversación en un sistema de diálogo hablado es muy parecido a la comunicación hablada de forma natural entre humanos. Necesitamos escuchar a la otra persona, para poder analizar la información que nos está proporcionando, y a partir de esa información, poder generar una respuesta que sea coherente con la conversación que estamos teniendo en ese momento.

### **2.1.1. Precedentes de los asistentes conversacionales.**

La historia de los asistentes conversacionales se remonta a los años 60, momento en el que fue desarrollado por Joseph Weizenbaum, el que es considerado como primer asistente conversacional de la historia, ELIZA. Fue planteado como una psicoterapeuta con el objetivo de demostrar la superficialidad que existía en la comunicación entre los humanos y las máquinas en la época. Mediante el reconocimiento de ciertas palabras clave, realizaba preguntas predefinidas, incluyendo dicha palabra clave.

Años más tarde, en 1972, Kenneth Colby creó a Parry, *chatbot* que simulaba el comportamiento de una persona con esquizofrenia, con el objetivo de ofrecer práctica a sus estudiantes antes de que se enfrentaran a pacientes humanos.

En 1988, llegó Jabberwacky, capaz de simular la voz humana en un tono humorístico. Creado con una finalidad meramente de entretenimiento, tenía como fin el cambio de un sistema de comunicación escrito a uno hablado.

El Doctor Sbaitso fue creado en 1992 por *Creative Labs*. Fue programado para simular a un psicólogo que ayudaría a resolver problemas emocionales a sus pacientes. La gran

mayoría de sus preguntas eran "¿Por qué te sientes así?". Cuando recibía una respuesta que no era capaz de entender simplemente le decía al usuario: "Es e no es mi problema".

El siguiente hito en la historia de los asistentes conversacionales llegó con ALICE (*Artificial Linguistic Internet Computer Entity*). ALICE era capaz de utilizar el lenguaje natural, para mantener conversaciones más sofisticadas con los usuarios. Una de las características de este *bot* es que es *open-source*, de tal manera que cualquier desarrollador podía utilizarlo para crear sus propios asistentes basándose en ALICE. Pese a no tener ningún objetivo ni propósito como asistente, fue de gran utilidad para futuros *bots*, e incluso sirvió de inspiración para una gran cantidad de películas.

En 2001 apareció *Smarterchild*. Fue distribuido a través de grandes redes de SMS, como MSN. Fue usado por más de 30 millones de usuarios, y sirvió de base el asistente de Apple.

Probablemente el *bot* más conocido por el gran público vino de la mano de Apple en 2010 con Siri. Sus funciones eran diversas, que iban desde la programación de agendas, añadiendo reuniones o eventos al calendario personal, o búsquedas por internet acerca de dudas que le pudieran surgir al usuario en cualquier momento. Su integración en el sistema operativo de Apple, iOS, le otorgaba la posibilidad de interactuar con aplicaciones instaladas en el propio terminal.

Actualmente el desarrollo de los asistentes conversacionales se encuentra en auge, y todas las grandes compañías cuentan con la ayuda de uno de ellos en la atención al cliente.

### **2.1.2. Situación económica.**

Los *chatbots* son a día de hoy, una herramienta que está siendo implementada en más y más negocios y compañías, independientemente del sector económico al que pertenecen. Hay una serie de razones por las cuales los asistentes conversacionales son un gran aliado de las compañías:

- Capacidad de reducir costes operacionales. La inclusión de este tipo de sistemas, podría reducir los costes hasta en un 30 % según la revista Chatbot Magazine [6].
- Mejora del compromiso del cliente. Con el objetivo puesto en conseguir clientes fieles a la compañía, los *chatbots* proporcionan respuestas instantáneas, y relacionadas al problema o duda que tenga el cliente en ese determinado momento, reduciendo tanto tiempos de espera por parte del cliente, como malentendidos que puedan surgir con agentes.
- Capacidad de entender mejor al usuario. Los asistentes virtuales, aparte de poder conversar con usuarios, tienen la capacidad de recolectar información y datos de los usuarios, tanto a través de las conversaciones mantenidas como mediante el uso de encuestas.

- Apertura de puertas a nivel internacional. La posibilidad de que los asistentes sean capaces de mantener conversaciones en diferentes idiomas puede resultar altamente útil cuando se trata de una compañía que opera en diferentes países o continentes. De este modo, se podrían evitar situaciones donde el idioma es una barrera para el cliente.

Los sectores donde la implementación de estos sistemas sería más interesante para reducir los costes operacionales son el de la venta al por menor, comercio electrónico, Banca y sanidad. En el año 2017 se esperaba una reducción de costes de 20 Millones de dólares, cifra que se espera que aumente hasta los 8 mil millones por año en 2022, según una investigación llevada a cabo por Juniper Research en mayo de 2017 [7]. Esta misma investigación sugiere que los bancos y las compañías de seguros sanitarios, podrían reducir entre 0,50\$ y 0,70\$ por interacción, partiendo de una interacción media por usuario de 4 minutos. Se espera que el nivel de interacción pase del 12 % al 75 % en 2022 en el sector de la sanidad, y llegue al 90 % en el sector bancario.

### 2.1.3. Situación actual.

El desarrollo actual en asistentes conversacionales para el gran público, se está centrande en dispositivos inteligentes que sean capaces de interactuar para realizar tareas relacionadas con la domótica y ciertas tareas más generales.

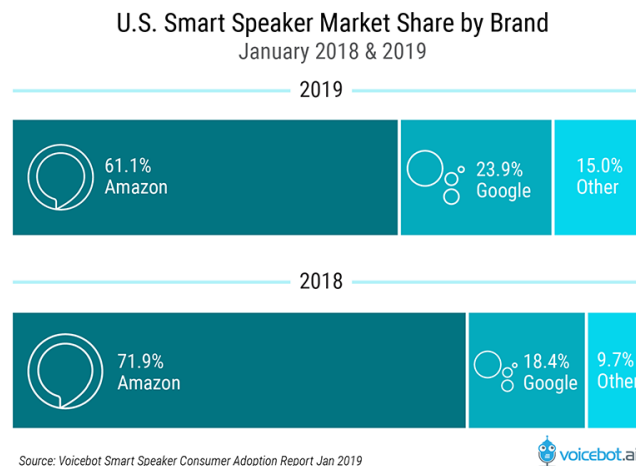


Fig. 2.1. Cuota de mercado de altavoces inteligentes [8].

Como podemos observar en la figura 2.1, son principalmente dos compañías las que lideran las cuotas de mercado en cuanto a ventas de altavoces inteligentes: Amazon y Google. Desde que el primer dispositivo de Amazon, el Amazon Echo, fuera lanzado en noviembre de 2014, multitud de dispositivos han sido lanzados con el paso de los años [9]. Amazon fue el ultimo en crear un asistente virtual, ya que tanto Apple, como Google como Microsoft ya habían desarrollado su propio *bot*. Sin embargo, fue pionero en incluirlo en un altavoz, idea a la que más adelante se sumarían Google y Apple. Las

funcionalidades presentadas por este tipo de altavoces son muy parecidas en todas las marcas, situándose las mayores diferencias en el diseño de los altavoces. Comparando las funcionalidades de los dispositivos de Google y Amazon, encontramos una alta cantidad de similitudes en su funcionamiento: ambos disponen de escucha continua, y despertarán con un comando de voz. El usuario les hace una gran variedad de consultas, que pueden ir desde añadir una alarma, añadir un elemento a una lista, poner música, o en el caso de que dispongamos de una casa con elementos conectados al altavoz, podremos pedirle que apague o encienda las luces, modifique la temperatura del hogar, o se encargue de mover las persianas.

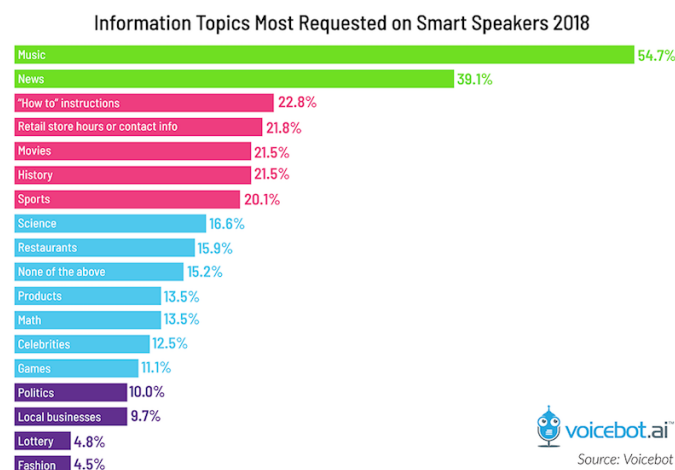


Fig. 2.2. Preguntas más frecuentes a los altavoces inteligentes[10]

## AMAZON ECHO [11]

El altavoz inteligente de Amazon, lanzado por primera vez en 2014, ha sido sustituido ya por su sucesor, la segunda versión del Amazon Echo. Mediante una petición a Alexa, el asistente conversacional de Amazon integrado en estos dispositivos, es capaz de poner música en cualquier lugar de la casa. Sus nuevas prestaciones incluyen encender la cafetera, cerrar las puertas, entre otras. Es capaz de conectarse con más de diez mil dispositivos diferentes para realizar tareas de forma automática en nuestro hogar [12].

El desarrollo en los últimos años de los campos del *aprendizaje profundo* (*deep learning*) y *aprendizaje automático* (*machine learning*) [13], permiten a Alexa ser más inteligente con cada uso y cada pregunta que le haga el usuario. Será capaz de anticiparse a algunas de nuestras preguntas, e irá aprendiendo nuestras expresiones y vocabulario para poder reducir el índice de malentendidos al mínimo.

Aparte de este dispositivo, Amazon ha lanzado dos nuevos dispositivos, ampliando así su gama de altavoces. El Amazon Echo Dot, un altavoz más pequeño, que podríamos catalogar de gama baja; en lo que podríamos catalogar como gama alta, encontramos el Amazon Echo Plus. Las funcionalidades que incluyen los diferentes altavoces son las mismas en los tres modelos, encontrándose las diferencias en el precio, la forma y colores de los altavoces y la calidad del sonido.

## GOOGLE HOME [14]

Por su parte Google lanzó su primer dispositivo en noviembre de 2016, dos años más tarde que el primer dispositivo de Amazon. Incluye como asistente al conocido Google Assistant, incluido también en todos los teléfonos Android.

Las funcionalidades que presenta Google en sus altavoces son idénticas a las que puede ofrecer Amazon, siendo la única diferencia la integración en el ecosistema Google para crear eventos en el propio calendario o el uso de otras aplicaciones nativas de Android.

A su vez, Google también dispone de varios dispositivos que cubren las necesidades de cada una de las gamas en las que se encuentran los de Amazon, para poder competir en todos los sectores. En la gama más baja, encontramos Google Home Mini, en la gama media Google Home y en la gama alta Home Max.

Pese a la importancia que están tomando en nuestra sociedad este tipo de altavoces que instalamos en nuestros hogares, las previsiones del mercado sugieren un cambio en esta tendencia creciente. Las tendencias, y los objetivos de las grandes empresas, no se centran en el desarrollo en exclusiva de asistentes de voz integrados en altavoces propios de la marca, si no en la inclusión de este tipo de tecnología en una gran variedad de dispositivos diferentes.

### 2.1.4. Ejemplos de asistentes conversacionales en el sector público.

Existen numerosos ejemplos de *chatbots* implementados por compañías privadas, pero los números en el sector público no son tan abundantes. La agencia Travel for London, creó en 2017 el *chatbot* TravelBot, para ayudar a cualquier usuario de la red de transporte público con cualquier consulta sobre rutas y horarios. La ciudad de Kansas incluyó en su página de datos abiertos un *chatbot* para guiar al usuario a través de la página. El gobierno de Dubai lanzó Rammas, capaz de responder a dudas y reclamaciones relacionadas con agua y la energía. Un ejemplo curioso es Mrs. Landingham, un bot creado para la agencia federal de los Estados Unidos, con el objetivo de ayudar a los trabajadores recién incorporados en trámites internos y rellenado de formularios [15].

En nuestro país, RENFE dispone IRENE [16]. El principal objetivo de este asistente es el de informar acerca de trayectos, horarios, empleo o como contactar con RENFE. Pero, sus funciones no acaban ahí, en su base de datos dispone de información como el tiempo, la hora en el mundo, direcciones de ciertas ciudades españolas, traducciones o significado de palabras entre otros.

### 2.1.5. Servicios web para el desarrollo de asistentes. Dialogflow.

Los asistentes conversacionales pueden ser creados desde cero, mediante la programación total de su lógica, su inteligencia artificial y utilidades. Existen diversas alternativas muy potentes para la creación de asistentes. La mayoría de estos servicios tienen en co-

mún su estructura y el uso del lenguaje natural interno para la programación [17]. El uso del lenguaje natural para el desarrollo de un asistente virtual tiene varias ventajas, como que cualquier usuario sin grandes conocimientos sobre la programación podrá crear un *chatbot* sin la necesidad de aprender un lenguaje para ello, y la que es la mayor ventaja, la posibilidad de acercar más al desarrollador al usuario final, permitiendo asistentes más intuitivos, simples y sencillos de usar.

Dialogflow, anteriormente conocida como API.AI, es uno de los servicios más conocidos para el desarrollo de asistentes conversacionales [18]. Bajo el mando de Google desde 2016, la compañía es también conocida por la creación de Assistant, el asistente nativo en Android.

Utilidades	Gratis	Edicion de pago	
		Essential	Plus
Conectores de inteligencia	Limitado	Limitado	Ilimitado
Texto o Google Assistant	Ilimitado	Ilimitado	Ilimitado
Audio	Limitado	Ilimitado	Ilimitado
Llamadas telefónicas	Limitado	Ilimitado	Ilimitado

Tabla 2.1. Planes de uso de Dialogflow [19]

Existen diferentes planes dentro de Dialogflow para el uso de sus servicios de creación de asistentes y de inteligencia virtual sostenida por Google.

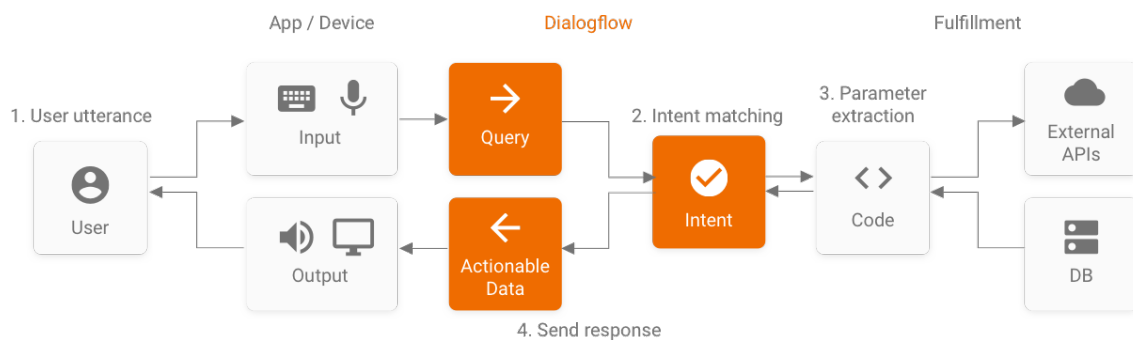


Fig. 2.3. Dialogflow: funcionamiento del fulfillment

En la Fig. 2.3 podemos observar como es el flujo de procesamiento interno del programa Dialogflow. En primer lugar necesitamos que el usuario despierte a nuestro asistente conversacional. Esto puede suceder mediante el inicio del asistente o una frase que le indique al asistente que sus servicios son requeridos por el usuario. Más adelante, el usuario le hará una petición al asistente. La información es enviada al API del agente, que realizará una búsqueda de palabras clave para poder responder algo coherente al usuario. Una vez encontradas dichas palabras clave, mediante el uso del lenguaje natural, el asistente busca entre sus diferentes *Intents*, intención del usuario, hasta encontrar uno que se adecúe a la petición. Este intent es capaz de seguir una lógica definida anteriormente por el



desarrollador, conectarse a una API externa o a una base de datos para proporcionar una respuesta al usuario. Este proceso es conocido en Dialogflow como *fulfillment*, o rellenado, y cada intent puede disponer de un fulfillment diferente [20]. Una vez la información que se le va a enviar al usuario es recogida, el asistente, mediante una serie de respuestas predefinidas, incluye dicha información y se la entrega al usuario mediante texto o voz [21].

Los intents o intenciones son las diferentes opciones de conversación que existen dentro de nuestro agente [22]. Los intents tienen cuatro componentes:

- Nombre del intent: el nombre es pasado al fulfillment para encontrar el intent que coincida.
- Frases de entrenamiento: posibles respuestas que puede dar el usuario. Dialogflow automáticamente expande estas frases mediante su inteligencia artificial para poder reconocer más posibles respuestas.
- Parámetro y acción: define como los parámetros son extraídos de la comunicación con el usuario. Estos parámetros pueden servir para buscar información, llevar a cabo una tarea o ser incluidos en la respuesta.
- Respuesta: lo que se le contesta al usuario tras su consulta.

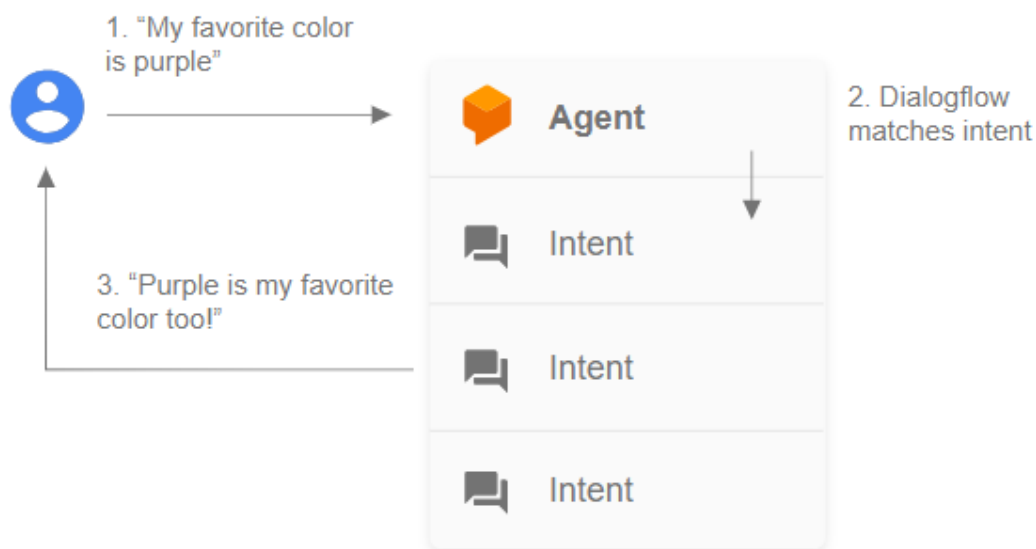


Fig. 2.4. Funcionamiento de los intents: ejemplo de las diferentes posibilidades de intents con los que cotejar la consulta del usuario.

Un agente dispone de varios *intents*, y cada vez que el usuario se comunica con nuestro agente, éste busca entre todos sus intents hasta encontrar el correcto. En la figura 2.4 podemos observar como existen diferentes *intents* que serán cotejados por el agente hasta encontrar el más parecido. Una vez dentro del intent adecuado, el agente enviará la respuesta al usuario. El método para asignar una consulta a un intent determinado se hace

mediante el uso de la detección de palabras clave y frases de entrenamiento, que le indican al agente posibles consultas que el usuario puede realizar. En el caso de que ninguno de los intents predefinidos por el desarrollador coincidan con la consulta del usuario, el agente acudirá al *fallback intent*. Este intent servirá como respuesta común ante una consulta no prevista o ante un fallo en la comprensión de la misma.

El siguiente elemento del agente son las *Entities* o entidades [23]. Son un mecanismo para identificar y extraer información desde el lenguaje natural usado por el usuario. Mientras que los intents tratan de entender la situación general de la conversación que se está teniendo con el usuario, las entidades tienen como objetivo la recolección de datos importantes que el usuario menciona. Toda la información relevante que se desee recoger de un intent, tendrá que tener su correspondiente entidad. Bajo un nombre común de entidad, se recogen todas las posibilidades que pueden existir bajo ese denominador común. Por ejemplo, bajo el intent colores, tendremos una lista completa de todos los colores que se deseen reconocer. Algo interesante que incorpora Dialogflow son las entidades del sistema, que por defecto serán capaces de reconocer fechas, lugares y momentos temporales introducidos por el usuario. El desarrollador del agente también podrá crear sus propias entidades, para que su agente lleve a cabo la labor propuesta.

## 2.2. Concepto de Smart Cities.

El concepto de ciudad inteligente o *Smart City* no es único [24]. Algunas definiciones se centran en el uso de las tecnologías de la información y comunicación (TIC) como el agente que mueve a las *smart cities* [25], mientras que otras definiciones más amplias incluyen aspectos socio-económicos, de sostenibilidad o la calidad de vida y el bienestar de los ciudadanos [26], [27].

En cualquier caso, el objetivo de una ciudad inteligente es el de mejorar la competitividad y asegurar un uso futuro mediante el trabajo conjunto de los ciudadanos, comercios, tecnologías, infraestructuras, consumo, energía y espacios[28].

La población en las ciudades está aumentando, y la mayor parte de la misma de la misma se concentra en las ciudades. Para el año 2050 se espera que la población mundial se doble en número, y que el 70 % viva en ciudades [1].

De esta manera, mediante las *smart cities* y sus iniciativas, se pretende mejorar la calidad de vida en las grandes ciudades, creando unas ciudades sostenibles medio ambientalmente y económicamente, mediante el uso de la tecnología. Así, se podrán mejorar los mecanismos de toma de decisiones, para asegurar una correcta interacción entre todos los componentes de las ciudades, asegurando la disponibilidad de los recursos para las futuras generaciones.

### 2.2.1. Características de las ciudades inteligentes.

Pese a que existen una gran cantidad de definiciones para las *smart cities*, y que en algunos casos estas definiciones se solapan con otros términos, como pueden ser: ciudad sostenible, ciudad digital o eco-ciudad.

Tampoco es posible crear un modelo de ciudad inteligente único y aplicable a todas las grandes ciudades del mundo, ya que influyen diversos factores tales como el tamaño de las ciudades o hasta el clima. Para poder englobar a la mayor parte de las ciudades en un único concepto común, y con una serie de características aplicables a todas ellas, se ha creado el proyecto de las ciudades inteligentes europeas (*European Smart City Project*).

El objetivo de este proyecto es el de lidiar con las perspectivas de crecimiento en tamaño, población y demanda de recursos de las ciudades medianas en Europa, dejando de lado a las grandes metrópolis. El principal motivo para llevar a cabo este proyecto, es que las grandes ciudades están siendo continuamente analizadas y disponen de una gran planificación de sus recursos presentes, y un detallado plan para el futuro. Sin embargo, las ciudades medianas, que también tienen que enfrentarse a los mismos problemas que las grandes ciudades, están menos equipadas en términos de recursos y capacidad de planificación y organización.

Este proyecto propone seis características comunes para las ciudades inteligentes:

- **Gobierno inteligente.** Por gobierno inteligente se entiende un gobierno dentro de la propia ciudad y entre ciudades que interactúe y tome decisiones conjuntamente, integrando organizaciones públicas, privadas, civiles y organizaciones de la Comunidad Europea, para que la ciudad funcione de un modo inteligente y efectivo. El principal medio para lograr este gobierno inteligente es el del uso de las tecnologías de la comunicación y de la información, siendo alimentado por datos.
- **Economía inteligente.** Engloba el comercio electrónico (*e-commerce*) y negocio electrónico (*e-business*), un aumento de la productividad. El uso de las tecnologías de la información y la comunicación será beneficioso a la hora de crear nuevos productos, servicios y modelos de negocios.
- **Movilidad Inteligente.** La movilidad inteligente se basa en una logística y transporte soportado en las TIC. Por ejemplo, diferentes sistemas de transporte tales como autobuses, tranvías, metros, trenes, coches y motos, sostenibles, seguros e interconectados entre sí. Este modelo de movilidad inteligente priorizará aquellos modos de transporte más limpios y libres de contaminación. El acceso a información en tiempo real resultará en unas conexiones más eficientes, permitiendo a su vez al usuario ahorrar tiempo al realizar estos trayectos[29].
- **Entorno Inteligente.** Se basa en la inclusión de las energías renovables, métodos de control de la contaminación, renovación de edificios, así como uso eficiente de

los recursos, reciclaje y reuso de los mismos.

- **Ciudadanos Inteligentes.** Este concepto hace referencia a las habilidades que los ciudadanos deben adquirir en relación a las TIC. Al vivir en un entorno dominado por información en tiempo real y datos, deberán ser capaces de utilizar esta información de un modo seguro e inteligente. Podrán utilizar toda esta información para sus propios propósitos o para la creación de empresas o servicios que a su vez ayuden al resto de ciudadanos.
- **Modo de vida Inteligente.** El modo de vida inteligente es básicamente un modo de vida en el que se premia el uso sostenible de los recursos y de la información disponible para los ciudadanos, que vivirán en ciudades multiculturales, conectadas y eficientes. Dispondrán de una gran cantidad de servicios y de una alta calidad de vida en todos los aspectos.

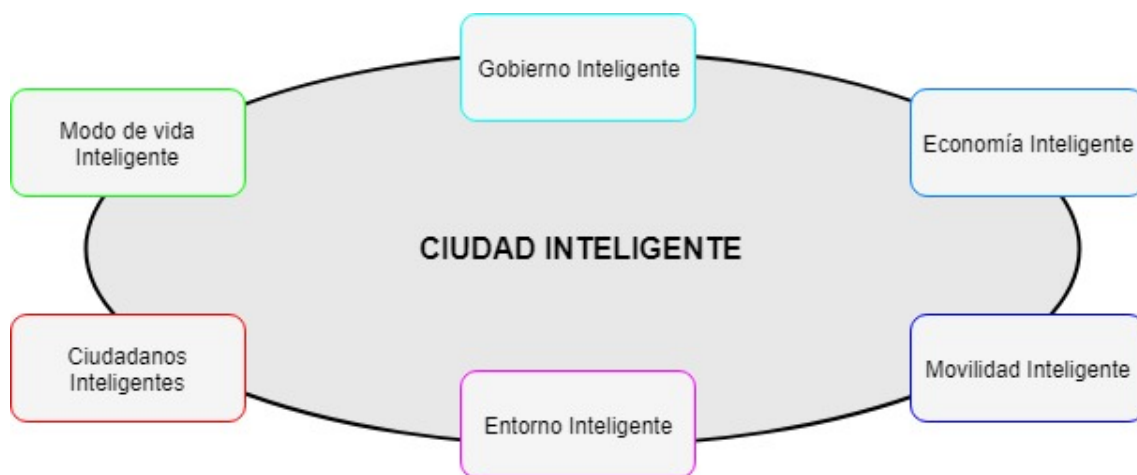


Fig. 2.5. Características de las Smart Cities. Las seis características comunes de las ciudades inteligentes [30]

La creación de servicios electrónicos y aplicaciones móviles, ayudaría enormemente en el objetivo de integrar transversalmente la gran mayoría de las características de una ciudad inteligente. La creación de portales con información abierta para el gran público acerca del funcionamiento de la ciudad, características de los edificios que la componen, e información sobre los diversos servicios que se ofrecen, es a su vez beneficioso para el fin de la creación de las *Smart Cities*.

La tecnología, junto con los factores humanos e institucionales son los tres componentes fundamentales sobre los que se basa el concepto de las ciudades inteligentes. Solamente, mediante la correcta fusión de estos tres factores, se podrá llegar a cumplir el objetivo planteado por la unión europea en materia de ciudades inteligentes.

La relación que existe entre los tres pilares y las seis características de las *Smart Cities* es tanto directa como indirecta.

### 2.2.2. Smart Cities y Europa 2020.

Los objetivos de las ciudades inteligentes están alineados con los objetivos de innovación y desarrollo planteados en Europa 2020.

"La Estrategia Europa 2020 es la agenda de crecimiento y empleo de la Unión Europea en esta década. Señala el crecimiento inteligente, sostenible e integrador como manera de superar las deficiencias estructurales de la economía europea, mejorar su competitividad y productividad y sustentar una economía social de mercado sostenible"[31].

Los principales objetivos los podemos encontrar en materia de empleo, I+D, cambio climático y energía, educación, pobreza y exclusión social.

Las iniciativas para las ciudades inteligentes pueden ser consideradas como una manera de conseguir los objetivos propuestos por Europa para el 2020. Los motivos para creer que estos objetivos están alineados y que el desarrollo en las *smart cities* ayudará a conseguir los otros objetivos es que las ciudades albergan a una gran cantidad de personas y consumen una gran cantidad de recursos. Producir un cambio estructural en las medianas y grandes urbes europeas supondría un gran impacto en los principales problemas, mucho más grande que si estos problemas se atacaran en pequeñas ciudades. La alta densidad de habitantes y de empresas, facilita el reconocimiento de problemas, movilización de la masa, y un reconocimiento y análisis de responsabilidades. Estos son algunos usos y características de las iniciativas para las ciudades inteligentes:

- El objetivo de Europa 2020 en energía (20 % de la energía deberá ser renovable, aumento del 20 % en la eficiencia energética) puede ser conseguido mediante iniciativas que se engloban dentro de la movilidad inteligente o entorno inteligente.
- Economía inteligente y ciudadanos inteligentes son iniciativas orientadas al desarrollo económico y la educación. Una correcta educación en temas tecnológicos resultaría en una mayor empleabilidad, atacando así el objetivo del empleo para 2020 (75 % de la población entre 20 y 64 años tendrá un empleo).
- Gobierno y ciudadanos inteligente ataca directamente los problemas en cuestiones de pobreza y exclusión social mediante un aumento de la calidad de vida, creando servicios para el beneficio de los ciudadanos.
- La mayoría de las medidas para conseguir una ciudad inteligente se basan en la modernización y la innovación, siendo un factor clave la inversión en I+D.

Existen otras maneras, más indirectas en como el desarrollo en una materia, a su vez resulta en el desarrollo en otra materia, que, a priori no está conectada con el primer objetivo.

### **2.2.3. Ciudades inteligentes y ciudades accesibles.**

Según informes de Naciones Unidas, en 2050, en torno al 15 % de la población mundial tienen algún tipo de discapacidad. Esta cifra aumenta hasta el 46 % en personas mayores de 60 años[32].

Dentro del contexto de las ciudades inteligentes, la movilidad inteligente y el entorno inteligente son dos características que tienen una gran implicación sobre la mejora de la accesibilidad en las ciudades. Con el avance de las tecnologías, se han ido implementando medidas en las urbes que ayudan a las personas con algún tipo de diversidad funcional a tener una vida más sencilla.

Existen muchos tipos de implementaciones, que podríamos catalogar como tecnologías activas y tecnologías pasivas. Dentro de las tecnologías activas, podríamos incluir todo el mobiliario de las ciudades que se han desarrollado para diversas funciones. Ejemplos de este tipo de tecnología son semáforos inteligentes o elementos que corten determinado tráfico en calles de paso restringido, que pueden ser accionados mediante un control remoto. Un elemento que ha sido de gran utilidad para la accesibilidad de todas las personas en las ciudades y ayudarles a realizar tareas tan sencillas como la de cruzar una calle es el semáforo que tiene avisos sonoros y un temporizador acerca de cuanto tiempo queda restante para poder cruzar. El aviso sonoro que se activa una vez el semáforo cambia a verde para los viandantes sirve de ayuda a personas con cualquier deficiencia visual, que normalmente no tendrían tan fácil el saber en que momento es correcto el cruce de la calle. Otros elementos que a día de hoy vemos como indispensables para la movilidad son los ascensores o las escaleras mecánicas para acceder a edificios, parquins o estaciones de tren y metro. El otro tipo de tecnología que nos encontramos en las calles de las urbes son las tecnologías pasivas. Estas tecnologías engloban suelos antideslizantes o rampas, que proporcionan una alternativa a aquellas personas que no puedan subir por las escaleras.

Mediante el uso de las tecnologías de la información y de la comunicación, crear ciudades inteligentes y accesibles está más al alcance de la mano. El uso de información para los ciudadanos, aportada por ciudadanos y organismos públicos, para su beneficio es el reto al que se enfrentan las ciudades ahora. Existe gran cantidad de información acerca de la accesibilidad de edificios públicos, estaciones de tren y metro, hospitales, oficinas de trabajo privadas; existe también información acerca de trayectos y rutas que son accesibles para personas con discapacidad, evitando medios de transporte o estaciones que causarían problemas. Toda esta información puede ser usada para crear mapas, aplicaciones o recursos para todos los ciudadanos, en un intento de crear una ciudad mucho más accesible para todos. Estas aplicaciones tendrían la posibilidad de ayudar con consultas sobre rutas más rápidas, eficientes, cortas o accesibles, según lo que sea preguntado. Añadiendo la posibilidad de acceso de la aplicación al GPS, podríamos ser seguidos en cada momento, y actualizar la ruta en tiempo real, en caso de que existiera algún retraso en algún medio de transporte, atasco o calle cortada.

#### 2.2.4. Proyecto SIMON

Simon es un proyecto llevado a cabo en cuatro ciudades diferentes: Madrid, Lisboa, Parma y Reading. Su principal objetivo es proporcionar más libertad e independencia a aquellas personas con movilidad reducida en el contexto de los medios de transporte mediante el uso de la información y otros sistemas. Este objetivo quiere ser alcanzado mediante la creación de una aplicación y una tarjeta europea de parquin para personas con discapacidad [33], [34].

La modernización de la tarjeta a una tarjeta contactless, permitiendo autenticación de su usuario, para facilitar el pago en los parquímetros o parquins. En cuanto a la aplicación tiene diferentes servicios aparte de información específica y funcionalidades de navegación; añade la posibilidad de utilizar el NFC del móvil para usarlo como tarjeta. Las funcionalidades de navegación incluyen la posibilidad de buscar rutas con requerimientos de accesibilidad en base a perfiles seleccionables, en vehículos privados, públicos o rutas peatonales.



Fig. 2.6. Captura de pantalla aplicación SIMON: distintas funcionalidades a las que se puede acceder desde la aplicación[34]

#### Retos a los que se enfrenta el proyecto

Los principales retos a los que se enfrenta este proyecto son varios. En primer lugar, desarrollar una aplicación móvil y una plataforma web con la utilidad de comunicarse con los ciudadanos para proporcionarles información; en segundo lugar, dar a las autoridades públicas la posibilidad de crear políticas inclusivas y luchar contra el fraude; en tercer lugar, crear servicios IT destinados a un público determinado, atendiendo siempre a la simpleza de la aplicación y facilidad de uso; en ultimo lugar, dotar a las personas con problemas de movilidad con las mismas oportunidades que el resto de ciudadanos.

## **Objetivos del proyecto**

El proyecto se encuentra en fase piloto, y cualquier usuario puede probar la aplicación móvil; cualquier persona con movilidad reducida que disponga de una tarjeta de estacionamiento podrá participar en las pruebas de la nueva tarjeta. Pese a que aún se encuentra en fase de pruebas, tiene una serie de objetivos que pretende cumplir:

- Mejorar la experiencia de usuario de las aplicaciones móviles para ciudadanos de la tercera edad y con movilidad reducida, mediante el uso de las TIC.
- Promover la accesibilidad mediante incentivos para los ciudadanos.
- Proteger la accesibilidad mediante un sistema de multas. SIMON pretende luchar contra el fraude y el mal uso de las tarjetas de estacionamiento de personas con movilidad reducida.
- Probar el proyecto en cuatro grandes ciudades para que sirva de base y sea extensible a otras urbes europeas.
- Reducir el coste operacional de la implementación de estos servicios mediante el sistema de multas e incentivos.

### **2.3. Sistemas operativos móviles.**

En los últimos años, los dispositivos móviles han ido cogiendo más fuerza en la sociedad, convirtiéndose en un aparato electrónico indispensable en el día a día de la gran mayoría de las personas.

Pese a que sus funcionalidades iniciales se limitaban a realizar llamadas, poco a poco, con el desarrollo de los diferentes sistemas operativos, estas funcionalidades han ido ampliándose, quedando las llamadas en un segundo plano. El salto ocurrió cuando aparecieron en escena los conocidos "smartphones."º teléfono inteligente. Posee características parecidas a las de un ordenador personal, pero en un tamaño más reducido.

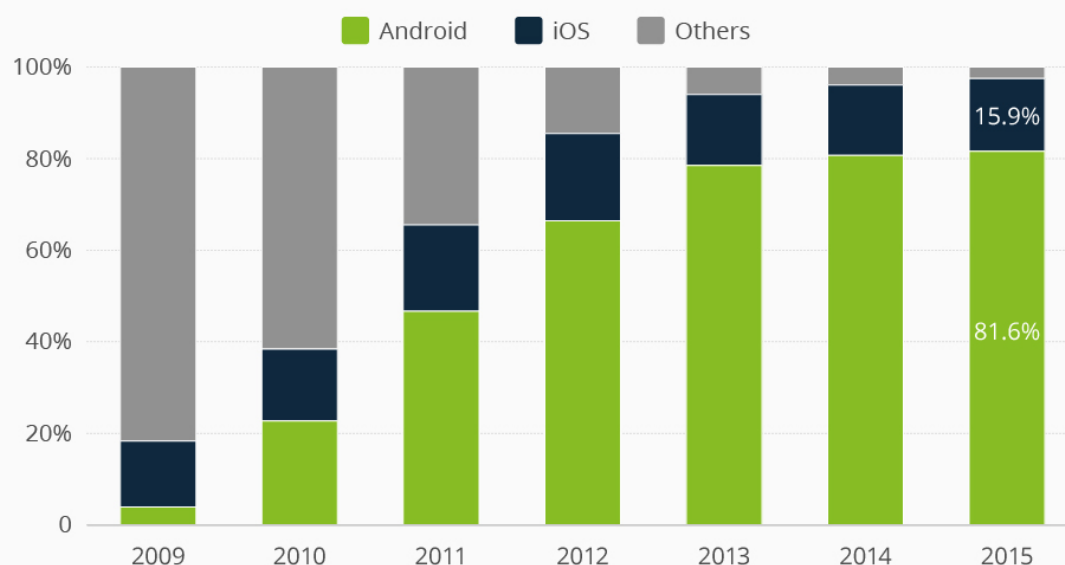
#### **2.3.1. Principales sistemas operativos.**

Los principales sistemas operativos incluidos en los diferentes terminales son dos: Android e iOS. Google es el dueño y desarrollador del primer sistema operativo, mientras que Apple tiene el control sobre iOS, estando presente tan solo en los dispositivos fabricados por la propia marca.



## Android and iOS Are the Last Two Standing

Worldwide smartphone operating system market share (based on unit sales)



@StatistaCharts Source: Gartner

statista

Fig. 2.7. Cuota de mercado Sistemas operativos. Evolución en los últimos años

- Android.** Es el líder del mercado en sistemas operativos para smartphones. Su origen se remonta al 2003. Fue creado por Android Inc. que sería más tarde adquirida por Google. Pese a ser concebido como un sistema operativo para móviles, su presencia no se queda solo en este tipos de dispositivos, si no que ha sido llevado a otros formatos como tablets e incluso PCs. Una de las características más importantes de este sistema operativo es que se desarrolla de forma abierta, de modo que cualquier desarrollador puede realizar modificaciones. A su vez, esto permite que una gran cantidad de fabricantes de teléfonos móviles utilicen Android como su sistema operativo base, sobre el cuál simplemente introducirán mejoras o cambios menores, que los diferencien del resto de sus competidores. Uno de los problemas viene con las actualizaciones del sistema operativo. Pese a que Google lance una nueva versión del sistema operativo base, será cuestión del fabricante el implementar esos cambios a sus dispositivos. Esto, en muchas ocasiones no sucede y los dispositivos quedan rápidamente obsoletos. Dentro de las características comunes de la gran mayoría de dispositivos Android es la inclusión de una tienda de aplicaciones: Google Play. En esta aplicación el usuario encuentra una gran variedad de aplicaciones que podrá instalar en su propio terminal.
- iOS.** Sistema operativo nativo para la gran mayoría de dispositivos Apple, tanto iPhones como sus tablets, más conocidas como iPad. Es considerado para el gran público como un sistema operativo simple e intuitivo, y dispone de una interfaz amigable para cualquier tipo de usuario. Uno de los mayores inconvenientes que

trae este sistema operativo es que es un sistema cerrado, lo cuál reduce las posibilidades de personalización por parte del usuario, así como la posibilidad de conexión con ciertos dispositivos. La tienda de aplicaciones nativa para estos dispositivos es Apple Store.

Otros sistemas operativos han estado presentes en la evolución de los smartphones, jugando incluso papeles muy relevantes para la historia de estos dispositivos. Entre estos sistemas operativos, que pese a haber sido de gran relevancia, han perdido presencia en el mercado actual:

- **Windows Mobile.** Este sistema operativo fue creado y desarrollado por Microsoft, para los precursores de los smartphones, donde no tuvo un gran éxito y quedó obsoleto pronto. Sufrió un auge con la llegada de los smartphones, pero nunca se llegó a situar como uno de los sistemas operativos preferidos por los usuarios. Los motivos para esto fueron principalmente la escasez de aplicaciones desarrolladas u optimizadas para estos dispositivos. La tienda de aplicaciones para este sistema operativo es Windows Marketplace.
- **BlackBerry OS.** Sistema operativo desarrollado por Research in Motion para los dispositivos BlackBerry. La gran diferencia de este sistema operativo es su facilidad para multitarea y experiencia de usuario para este tipo de dispositivos. Una de las características que hacía únicos a estos dispositivos es la inclusión de un teclado físico QWERTY junto con una rueda para moverse dentro de la aplicación actual, así como entre aplicaciones. Pese a ser un ecosistema muy enfocado a la productividad, ha sido desplazado por nuevos smartphones.

### 2.3.2. Arquitectura de Android.

Android es una plataforma que sigue una arquitectura en forma de pila, distribuida en diferentes niveles, donde el nivel superior se apoya en los niveles inferiores y sirve a su vez de base para los niveles superiores, es decir, se va construyendo por capas [35].

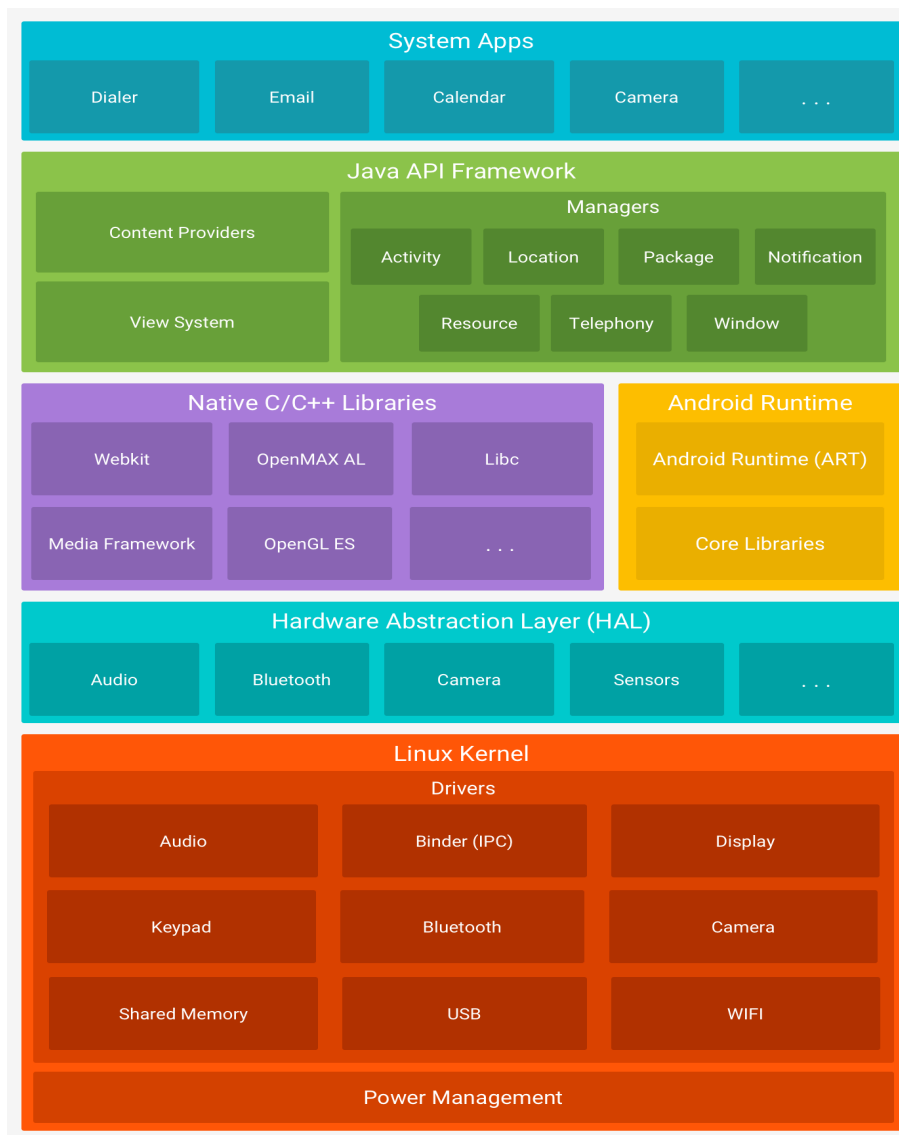


Fig. 2.8. Arquitectura android[36]

- **Kernel de Linux.** Es la base de la plataforma Android. Es la primera capa de software, que funciona como intermediario entre los componentes de hardware y software. Contiene los drivers que controlan cualquier componente de hardware. Para que los componentes de hardware puedan ser utilizados, es necesario incluir en esta capa base las librerías o drivers necesarios para que las llamadas a dichos componentes se realicen correctamente. Por ejemplo, el tiempo de ejecución de Android (ART) se basa en el Kernel de Linux para diversas funcionalidades[37].
- **Capa de abstracción de hardware (HAL).** Es una interfaz entre el software y hardware del sistema. Las aplicaciones, en lugar de acceder directamente al hardware, acceden a esta nivel. Esta capa de abstracción trae unas interfaces estándar donde se muestran las capacidades del hardware a los niveles superiores. Está compuesto por módulos de bibliotecas, donde cada componente de hardware incorpora una interfaz propia.

- Tiempo de ejecución de Android. Todas las aplicaciones son compiladas enteramente cuando son instaladas en el dispositivo. Esta implementación permite un rendimiento mucho mas alto. Como contra encontramos que las aplicaciones instaladas en nuestro dispositivo van gastando recursos del mismo, ocupando un espacio de memoria dentro del mismo. Este cambio en la forma de compilar las aplicaciones vino de forma nativa a partir de la versión Android 5.0. Antes de esta versión se utilizaba Dalvik como tiempo de ejecución del sistema. Dentro de las funciones de ART, encontramos:
  - Compilación ahead-of-time o compilación antes de tiempo (AOT), y just-in-time o justo a tiempo (JIT)
  - Mejor compatibilidad con la depuración.
  - Recolección de elementos no usados optimizada.
- Bibliotecas C/C++ nativas. Esta capa de librerías utilizadas por Android, han sido desarrolladas con C o C++, y proporcionan al sistema operativo la mayor parte de sus capacidades. Entre las librerías más importantes podemos encontrar:
  - Librería libc. Incluye cabeceras y funciones en lenguaje C. El resto de librerías son definidas en este lenguaje.
  - Surface Manager. Encargada de la composición de las pantallas y los diferentes elementos de las ventanas de las aplicaciones activas en cada momento.
  - OpenGL/SL y SGL. Son librerías gráficas, y por tanto, encargadas de los gráficos en 2D y 3D.
  - WebKit. Es el motor tanto del navegador incluido de forma nativa en Android como de las aplicaciones del tipo navegador.
  - Media Libraries. Incluye los códecs para el contenido multimedia.
  - SQL Lite. Biblioteca destinada a la creación y gestión de bases de datos relacionales.
- Framework de la Java API. El conjunto de funciones del sistema operativo se basa en las API escritas en lenguaje Java. Sirven para la creación de aplicaciones. Los desarrolladores de aplicaciones tienen acceso a las mismas API que utilizan las aplicaciones del sistema. Las API más relevantes son:
  - Sistema de vista para compilar la interfaz del usuario de la aplicación, donde se incluyen diferentes elementos como botones, cuadros de texto o un navegador web integrable.
  - Administrador de recursos, que gestiona los elementos que están dentro de la aplicación pero que se encuentran fuera del código, como gráficos o archivos de diseño.

- Administrador de notificaciones que permite el envío de alertas personalizadas para cada aplicación.
  - Administrador de actividad que maneja el ciclo de vida de las aplicaciones y proporciona un sistema de navegación entre ellas.
  - Proveedores de contenido, que permite a las aplicaciones acceder a información recogida en otras aplicaciones.
- **Aplicaciones.** La última capa del sistema operativo es la que alberga las diferentes aplicaciones, tanto de sistema como las que el propio usuario pueda instalar en su terminal. Las aplicaciones instaladas por el usuario tienen la capacidad de sustituir a las aplicaciones predeterminadas del sistema, ya que las aplicaciones incluidas en la plataforma no tienen ningún estado especial entre las apps que el usuario decida instalar.

### 2.3.3. Arquitectura iOS.

La estructura de este sistema operativo es parecida a la arquitectura de Android en términos de la distribución, teniendo ambas una forma de pila, dónde las capas superiores tienen su base sobre las inferiores y sirven de base para las superiores [38], [39].

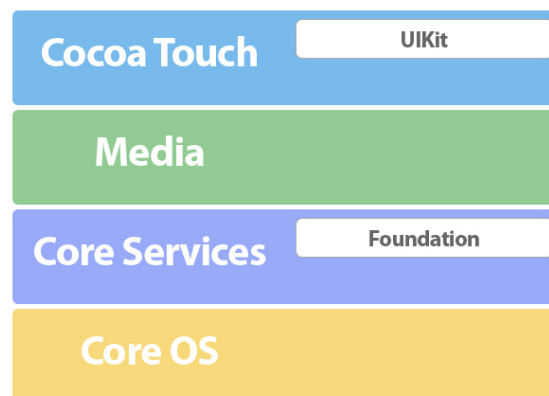


Fig. 2.9. Arquitectura iOS[40]

En este caso, nos encontramos con cuatro diferentes capas:

- **Core OS.** La mayoría de las funcionalidades de las tres capas superiores están construidas sobre esta capa. Esta capa incluye una serie de *Frameworks* que las aplicaciones podrán usar de forma directa. El sistema operativo iOS está basado en el OS X de Apple, desarrollado a partir de Unix.
- **Core services.** Esta capa se encarga del manejo de los principales servicios del sistema que utilizan las aplicaciones nativas de iOS. A su vez, proporciona una serie de características indispensables como compras In-App o almacenamiento en iCloud,

el servicio de almacenamiento en la nube de Apple. Como *Framework* principal encontramos Core Foundation, que permite el intercambio de datos y código entre las librerías y los *Frameworks*.

- **Media.** Los gráficos, el audio y el vídeo son manejados por esta capa. Es una interfaz con un lenguaje que es una mezcla entre C y objective C, que permite ejecutar tareas o gestionar archivos multimedia. Los principales *Frameworks* son:
  - Core Animation. Orientado a la animación de las aplicaciones para una mejor experiencia de usuario.
  - Media Player Framework. Es un *Framework* de alto nivel destinado a la reproducción de musica y *playlists* en iTunes.
  - AV Kit. Destinado a la reproducción de vídeo.
- **Cocoa Touch.** Es la capa de nivel superior de la arquitectura iOS. Es la capa que los usuarios utilizan para interactuar con las aplicaciones. Contiene el *Framework* UIKit, que sirve de base para animar el contenido de la pantalla, desde la multitarea hasta la gestión de la entrada táctil por pantalla [41]. Es el *Framework* en el que se basan la gran mayoría de aplicaciones nativas de iOS. Esta capa contiene características tales como el reconocimiento de gestos o el soporte para la visualización de documentos.

### **3. DISEÑO Y DESARROLLO DEL SISTEMA.**

En este capítulo se presentan los diferentes componentes del proyecto, tanto el agente conversacional como la aplicación en la que se encontrará el asistente, y sus diferentes funcionalidades. Se explican las herramientas que se han empleado para desarrollar los diferentes sistemas. El desarrollo termina con una serie de pruebas llevadas a cabo por el programador y una evaluación llevada a cabo por usuarios finales de la aplicación. Esto se detallará al final del capítulo, presentando las pruebas a las que se somete el sistema así como los resultados esperados y obtenidos.

#### **3.1. Descripción de las funcionalidades básicas.**

Este proyecto, viene a cubrir una de las necesidades básicas que tienen las personas con algún tipo de diversidad disfuncional en la comunidad de Madrid. El objetivo del proyecto es crear una aplicación capaz de ayudar con la movilidad y la accesibilidad en toda la comunidad de Madrid, tanto en estaciones de transporte público, como en edificios de carácter municipal o no municipal.

La aplicación en cuestión, permite al usuario tener una conversación en lenguaje natural con un asistente conversacional, al que le podrá hacer preguntas acerca de la accesibilidad de una determinada estación u edificio. La conversación se podrá realizar tanto por texto como por voz, teniendo también ambas posibilidades el asistente. Dependiendo del medio que utilice el usuario para hacer la consulta, el asistente responderá a través del mismo medio, facilitando así una conversación más natural. Tener la posibilidad de escribir o hablar, y de poder leer o escuchar, permite que la aplicación en sí, sea accesible a un gran público, y facilita la tarea a la hora de encontrar información sobre la accesibilidad. Por tanto, la aplicación debe ser intuitiva y facilitar la interacción en todo momento.

Una vez definido esto, podemos ver cuál es la principal función de la aplicación: permitir al usuario establecer una conversación con un asistente virtual que le ayudará a conocer la accesibilidad allí donde desee. La aplicación en sí no es más que un simple medio para poder realizar la comunicación entre ambas partes, pretendiendo simular un entorno que el usuario sienta como familiar. La aplicación, por tanto, es simplemente una interfaz que facilita el uso del asistente, y le dota con una serie de funcionalidades que pueden resultar ciertamente útiles al usuario. La aplicación mostrará la interacción entre ambas partes, siendo el agente que se encuentra por detrás, el encargado de que esa conversación se realice satisfactoriamente. Este proyecto, cuenta así, de varias partes que están interconectadas, y necesitan de la otra para poder funcionar correctamente: un agente conversacional, y una aplicación, que albergue dicho asistente.

Para que la utilidad de la aplicación sea máxima, debe tener las siguientes capacidades:

- Permitir la entrada por texto por parte del usuario. La aplicación contará con un campo donde el usuario podrá escribir su consulta o responder a las preguntas que el asistente vaya realizando en las diferentes iteraciones de la conversación, hasta llegar a la consulta final.
- Permitir la entrada por voz por parte del usuario. En caso de que para el usuario sea más cómoda o efectiva la comunicación por voz, podrá comunicarse por dicho medio con el asistente. Mediante el módulo de *Speech-to-Text*, la aplicación transformará las consultas realizadas por voz a texto, para que así el agente pueda analizar las respuestas del usuario.
- Manejar las respuestas del asistente. El usuario, cada vez que hable con el asistente, recibirá una respuesta por parte de éste. La respuesta llegará a la aplicación desde el agente conversacional en forma de texto. Es tarea de la aplicación analizar el medio que está usando el usuario y presentarle la respuesta por el mismo medio por el que se está comunicando el usuario. En caso de que el usuario se comunique por voz, la aplicación, mediante el módulo *Text-to-Speech*, hablará la respuesta que le ha llegado en forma de texto desde el agente.
- Permitir al usuario localizar el lugar de la consulta. El usuario, una vez haya recibido la información sobre si el edificio o estación son accesibles, y en caso de que la respuesta sea positiva, tendrá la posibilidad de localizar en un mapa con máxima precisión donde se ubica dicho edificio o estación. De esta manera, el usuario, que desconoce el estado de la accesibilidad, puede fácilmente recibir instrucciones acerca de cómo llegar hasta dicho punto.
- Permitir al usuario tener un listado con sus lugares favoritos. El usuario que lo desee, después de conocer la accesibilidad de la infraestructura consultada, podrá guardarla en su pestaña de lugares favoritos, para poder acceder a la información relacionada a dicho punto en cualquier momento, tanto sobre la accesibilidad como su localización.
- Recibir alertas en tiempo real que se puedan lanzar acerca del estado específico de la accesibilidad en un determinado edificio o estación.

La última de las características que tiene la aplicación es una de las funcionalidades más relevantes que implementa la aplicación es la posibilidad de recibir alertas y notificaciones en tiempo real acerca del estado de las instalaciones en los diferentes puntos de interés. Esto es realmente útil ya que, el usuario que dependa de las instalaciones de accesibilidad para acceder a las estaciones o edificios necesita conocer si están en funcionamiento. No es extraño ver ascensores averiados, escaleras mecánicas paradas, y otros elementos en mal estado. Estos problemas, que para el usuario de transporte público normal pueden pasar desapercibidos o no marcan ninguna diferencia, son de gran importancia para aquellos usuarios con diversidad disfuncional y que necesitan dichos elementos para poder acceder o salir de las instalaciones. El poder conocer a tiempo real esta información, y recibir



notificaciones *push*, para que hasta desde fuera de la propia app, el usuario sea consciente de si existe alguna avería o imprevisto, para así, poder coger otra línea de metro, bajar en otra estación o incluso ir a dicho edificio otro día o en otro momento.

Por otro lado, es importante conocer cuáles son las funcionalidades de las que dispone el agente conversacional. Podríamos decir que el agente es el motor que mueve la aplicación, y que va guiando al usuario a través de preguntas a que realice una consulta específica y determinada, con una serie de parámetros, que serán los que servirán para hacer la consulta a las API. Este tipo de funcionalidades no están implícitas en el asistente en sí, si no que mediante conexiones a diferentes servidores y API podrá realizar dichas consultas y recabar la información consultada por el usuario. Las principales funcionalidades son:

- Modelo de agente orientado al proyecto. Pese a que la inteligencia artificial de la que dispone el agente conversacional ya está programada anteriormente por parte de Google, el dueño del programa con el que se realiza el agente; es tarea del desarrollador del agente el orientar dicho agente en la dirección en la que se desee. Por tanto, mediante frases de entrenamiento, intenciones y entidades, se modelará el agente para que sea capaz de identificar la intención del usuario, guiar la conversación y ser capaz de formar una respuesta coherente con la consulta del usuario y en el momento de la conversación.
- Identificar las entidades. A través del reconocimiento de *entities* o entidades, el agente podrá saber qué desea conocer el usuario, generar respuestas coherentes para la pregunta formulada y consultar las API para conseguir la información que complete la respuesta.
- Reconocer la intención del usuario. Ya que existen diferentes tipos de consultas que puede realizar el usuario, es importante que el asistente conversacional conozca cuál es la intención del usuario, para así poder proporcionarle las respuestas más adecuadas a sus preguntas.
- Generar respuestas y preguntas. Es importante que el agente sea capaz de generar tanto respuestas como preguntas al usuario para poder continuar con la conversación hasta que el usuario haya conseguido la información que necesita.

Es importante que la aplicación y el agente estén conectados para que las consultas que realice el usuario sean enviadas al agente, y para que las respuestas que genere el agente puedan ser proporcionadas al usuario.

### **3.2. Agente conversacional.**

El primer gran elemento que ha de ser desarrollado es el agente conversacional, ya que será el encargado de mantener la conversación con el usuario. El agente conversacional

será integrado en la aplicación y tendrá la capacidad de analizar la semántica y palabras clave que pueda usar el usuario para así elaborar respuestas.

La diferencia entre un agente conversacional y una búsqueda en un motor de búsqueda de resultados web reside en la manera en la que se formulan las preguntas. El realizar consultas a través de un agente conversacional permite al usuario realizar preguntas y consultas cómo las podría hacer a través de una llamada telefónica o incluso a otro humano. Esto facilita la comunicación de forma natural y en un lenguaje natural, lo cuál resulta mucho más cómodo. Uno de las características más importantes de los agentes conversacionales y que hacen que tengan cada vez más peso en nuestra sociedad es que disponen de una completa disponibilidad al usuario, y tienen siempre como objetivo el llegar a la consulta final para poder ayudar al usuario, sin tomar otros caminos ni perder el tiempo de usuario.

En este proyecto, se ha decidido utilizar el servicio de creación de asistentes conversacionales de Dialogflow, anteriormente presentado en el Estado del arte. Permite utilizar una gran cantidad de recursos y características en su versión gratuita, lo que le sitúa como una de las alternativas más interesantes. El nivel de desarrollo de la inteligencia artificial que tiene esta plataforma, así como la capacidad de ir aprendiendo según van avanzando las interacciones con diferentes usuarios hacen que las conversaciones que se tienen con agentes desarrollados con esta plataforma web sean de un gran realismo y de una elevada eficiencia a la hora de entender las cuestiones planteadas por el usuario y resolver sus dudas o realizar las tareas propuestas. Una gran cantidad de servicios de Google están conectados e incluidos en Dialogflow, tales como el módulo *Speech-to-text*.

Dialogflow permite con la misma cuenta de usuario crear diferentes agentes, pero para este proyecto necesitaremos crear tan solo uno. Será capaz de establecer una conversación básica con el usuario, donde le preguntará si desea conocer la accesibilidad en un edificio o en una estación, y más tarde le preguntará el nombre del edificio o estación en cuestión. Para poder desarrollar un agente que satisfaga estas necesidades tendremos que usar las intenciones o *intents* y las entidades o *entities*.

### **3.2.1. Conversación entre agente y usuario.**

Los agentes conversacionales creados mediante servicios web hacen uso de la inteligencia artificial y del aprendizaje automático y progresivo según los usuarios realizan más consultas e interacciones con los agentes. Esta información es mostrada igualmente al desarrollador para que pueda realizar análisis y evaluaciones de su propio agente para ver cuál es el nivel de compromiso del usuario, los principales fallos del agente a la hora de entender las intenciones o las palabras clave más utilizadas. Para poder crear las intenciones y las entidades es importante conocer cuáles van a ser los objetivos que tenga el agente y el flujo que se va a seguir en las conversaciones. En este caso, el flujo en que se va a producir la interacción entre el agente y el usuario será a través de la aplicación que será más tarde desarrollada, como podemos ver en la figura 3.1.

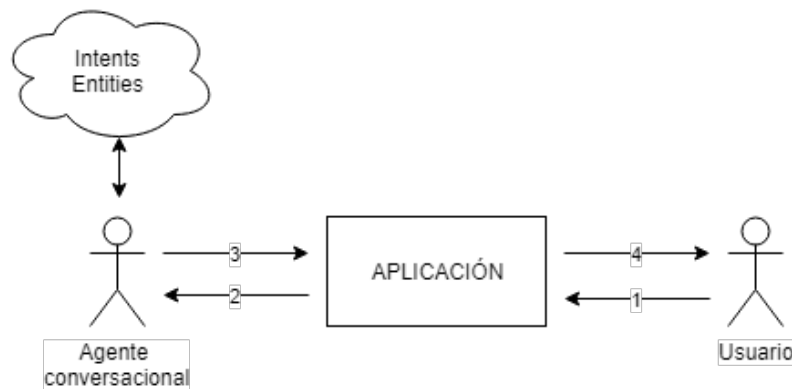


Fig. 3.1. Flujo de conversación entre el agente y el usuario

En la figura anterior podemos comprobar como existen diferentes elementos dentro de la conversación: el agente, la aplicación y el usuario. Interaccionan entre ellos en diferentes pasos:

1. El usuario realiza la primera interacción y es el encargado de llamar al agente para realizarle la consulta.
2. La aplicación le envía la información al agente para que éste pueda recoger los datos, analizarlos y elaborar una respuesta.
3. El agente envía la información a la aplicación que será la encargada de presentarle esta información al usuario mediante texto o voz.
4. La aplicación le presenta finalmente al usuario la respuesta que ha llegado desde el agente creado en Dialogflow.

Este flujo conversacional se repetirá tantas veces como sea necesario hasta que el usuario tenga respuestas a sus dudas. Por su parte, el agente le realizará tantas preguntas como sean necesarias al usuario para que así pueda comprender totalmente sus intenciones y reconocer las entidades o palabras clave con las que realizar las consultas a las bases de datos, servidores y API y poder responder coherentemente y con información personalizada al usuario con el que está realizando la conversación.

Las interacciones que tienen los agentes con los usuarios pueden ser de dos tipos: diálogos abiertos o diálogos dirigidos. El primer tipo de interacción, se basa en que el usuario puede realizar diferentes preguntas seguidas, de temas no necesariamente relacionados o conectados. Este sistema de interacción de diálogo abierto es el utilizado en asistentes integrados en terminales móviles como pueden ser Siri o Google Assistant. En estos casos, el usuario puede realizar una consulta a la que responderá el agente. Después, si desea realizar otra consulta diferente el agente se reinicia y está preparado para responder a cualquier otra cuestión. Por su parte, en el diálogo dirigido, el agente guía la conversación hacia un camino predefinido previamente por el desarrollador. El diálogo

en estos casos es más cerrado, y el agente no tiene tantas posibilidades de responder a preguntas seguidas de diferentes temas, si no que se centrará en responder una serie de cuestiones específicas.

En este proyecto, el agente creado utilizará un sistema de diálogo dirigido. El agente no está orientado a la simple conversación ni a la conversación de temas que no estén relacionados con la accesibilidad en edificios y estaciones enmarcadas en la Comunidad de Madrid. Será capaz de reconocer la intención del usuario en este ámbito, pero será incapaz de recomendar al usuario un restaurante o comentar cómo será el tiempo en Barcelona la próxima semana. El motivo principal por el que se ha pensado que sería más eficiente un agente de este tipo es debido a que el usuario que vaya a conversar con el agente es debido a que tiene un problema puntual o una cuestión específica sobre algún punto de interés en la comunidad de Madrid, y cuánto más sencillo le sea al usuario preguntar y encontrar una respuesta a su consulta, mayor será el compromiso que adquiera el usuario con el agente y existirá una mayor probabilidad de que lo vuelva a usar en futuras interacciones en lugar de una búsqueda online. Prácticamente ningún usuario tendrá descargada la aplicación en su terminal para conversar con el agente acerca de temas como el tiempo o restaurantes, ya que para estos temas tan diversos se dispone de los asistentes integrados nativamente en los sistemas operativos.

### **3.2.2. Intenciones o *intents*.**

Las intenciones son uno de los elementos principales a tener en cuenta a la hora de desarrollar un agente con Dialogflow. Si dividiéramos nuestro agente en diferentes funciones con las que sería capaz de ayudar al usuario, claramente distinguidas cada una de la otra, esas serían las intenciones de las que dispone el agente. Por tanto, crearemos tantas intenciones dentro de nuestro agente como cuestiones que queremos que comprenda y sea capaz de responder correctamente.

En este caso, ya que el usuario puede preguntar tanto por edificios como por estaciones, crearemos dos intenciones diferentes. Dentro de cada intención, introducimos lo que se conoce como frases de entrenamiento. El objetivo de estas frases de entrenamiento es preparar al agente ante posibles preguntas o respuestas que puede realizar el usuario. La inteligencia del agente no se limita a las frases de entrenamiento, y no será necesario que el usuario escriba lo mismo que está programado en las frases de entrenamiento, si no que el agente puede ir aprendiendo automáticamente sobre cómo los usuarios realizan las diferentes consultas. En el caso de que no entienda la pregunta del usuario, simplemente le pedirá que repita la consulta de nuevo, con otras palabras para así poder comprenderlo.

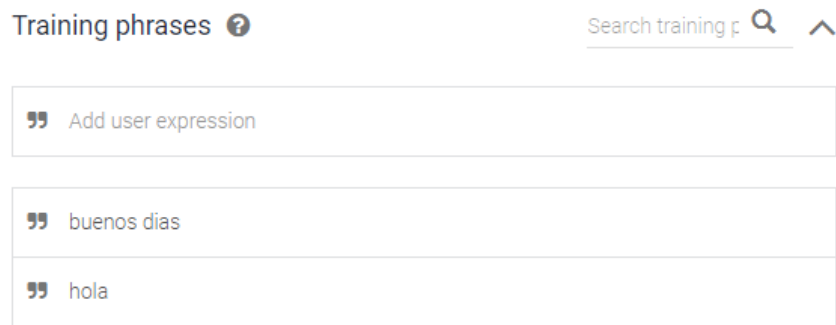


Fig. 3.2. Frases de entrenamiento

En la figura 3.2, se muestran las frases de entrenamiento que activarían al asistente conversacional. Cuando el usuario diga una de esas frases, hará que el asistente comience a funcionar y a escuchar las consultas del usuario.



Fig. 3.3. Frases de entrenamiento con parámetros.

Es importante que en las frases de entrenamiento se incluyan palabras clave que sirvan al agente a la hora de determinar cuál es la intención. Podríamos decir que las palabras clave son algo así como las diferentes variables que el usuario puede introducir en sus consultas. Estas variables son conocidas como entidades dentro de Dialogflow, y algunos ejemplos se muestran en la figura 3.3. En esta imagen podemos ver como el agente reconoce una serie de parámetros, y esperará que el usuario mencione alguno de esos parámetros al realizar su consulta. A su vez, estas variables son las que el agente utiliza para realizar las consultas a sus bases de datos o servidores externos y poder devolver información al usuario. Podemos indicarle a Dialogflow que algunas de estas variables son

necesarias para el correcto funcionamiento del agente. En el caso de que el usuario deje claro que cuál es su intención, pero no haya proporcionado información al agente acerca de cuál es la variable, el agente le preguntará directamente cuál es el valor de la variable que necesita.

Existen otros elementos dentro de las intenciones. Encontramos las respuestas que puede dar el agente dentro de las intenciones, y el contexto. El contexto es útil si se pretende mantener información de una intención a otra, para realizar consultas relacionadas entre sí.

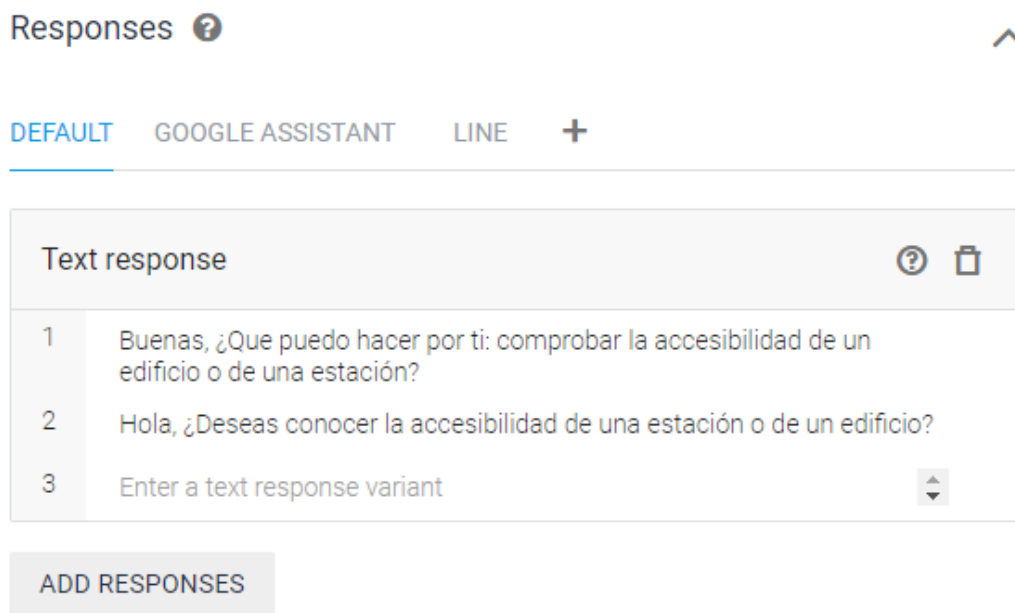


Fig. 3.4. Respuestas programadas del agente

Mediante una serie de respuestas programadas en cada intent, el agente podrá responder lo que queremos al usuario. Además de las respuestas programadas, existe la posibilidad de que el agente incluya en su respuesta ciertas palabras clave o parámetros que ha mencionado el usuario para así crear una respuesta más elaborada y personalizada. También es posible crear respuestas desde el *Webhook* utilizado para buscar información. Esta es la manera que tiene el agente para realizar comunicaciones con servidores externos y recoger información que le será útil al usuario y que por tanto ha de incluir en su respuesta. Además de incluir la información que ha sido incluida por el usuario en su pregunta, será de esta manera capaz de incluir nuevos datos. El *Webhook* está ligado a cada intent y cuando el intent sea activado, y los parámetros requeridos sean recogidos, se realizará la petición http con un JSON donde se incluyen dichos parámetros.

Continuando con la explicación del proyecto, tenemos dos intenciones creadas para el propio proyecto, más las dos intenciones que incluye de forma automática Dialogflow. La primera intención que incluye el programa es la conocida como intención de bienvenida. Esta intención es activada con la primera intervención del usuario, aunque puede ser

activada mediante otros métodos. Se ha elegido que el usuario sea el que inicie la conversación, ya que podría entrar en la app para ver sus lugares favoritos y no necesariamente para conversar con el asistente. Pese a que los asistentes son bastante inteligentes, y cada vez lo son más, no tienen una capacidad de entendimiento del cien por cien, y en algunas situaciones se producen malentendidos. El módulo de *Speech-to-Text* puede cometer algún error a la hora de interpretar lo que ha dicho el usuario, así como se da la posibilidad de que el usuario se exprese mal o por algún casual cometa algún fallo de ortografía o escritura que haga ininteligible la frase para el asistente. En estos casos donde la comunicación no es sencilla y la intención del usuario no se puede ver con claridad, el programa incluye la intención de error o *Fallback Intent*. Esta intención viene programada de serie con una serie de respuestas que el agente dirá en caso de que no comprenda lo que quiere el usuario. Existe también la posibilidad de que el usuario realice alguna consulta para la cuál el agente no esté preparado ni haya previsto el desarrollador. En este caso, también se activará esta intención.

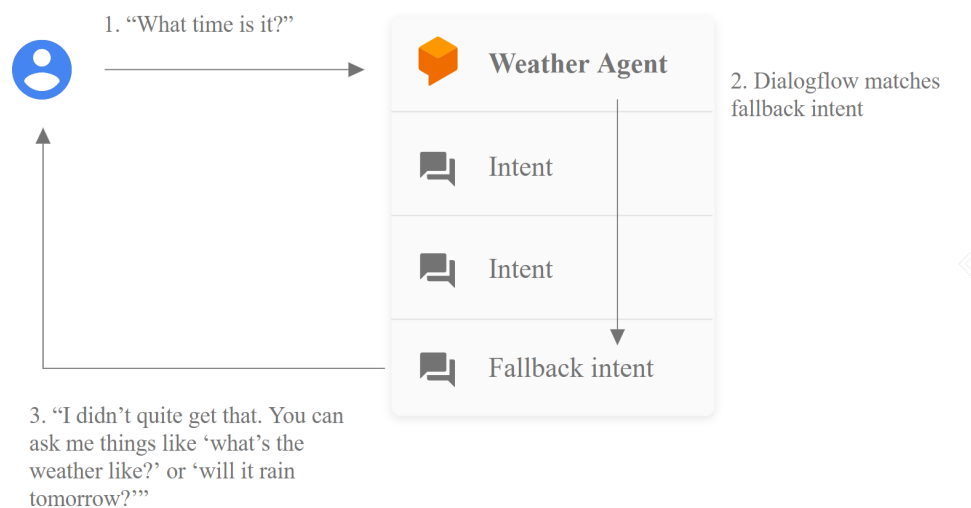


Fig. 3.5. Activación del *Fallback intent*.

Como observamos en la figura 3.5, el agente cotejará la intención del usuario entre los diferentes intents de los que dispone. En caso de que no encuentre ninguna coincidencia, activará el *fallback-intent*, y le dirá al usuario que no ha sido capaz de entenderle, que repita de nuevo la consulta.

Las intenciones creadas exclusivamente para este proyecto se tratan de los intents llamados "estaciones" y "edificios". El primer caso se activará cuando el usuario pida conocer la accesibilidad de una estación del transporte público de la Comunidad de Madrid, mientras que el segundo se activará cuando el usuario pida lo mismo pero para un edificio.

Dentro de las posibles respuestas que le puede dar el agente al usuario, existe la posibilidad de que éste incluya las *entities* que el usuario puede haber dicho durante su intervención en la conversación, para dar así una sensación de escucha mucho mayor.

### 3.2.3. Entidades o *Entities*.

Las variables que se mencionaban anteriormente, son conocidas como entidades o *entities*. Las entidades son utilizadas para extraer información relevante que el usuario menciona en sus diferentes intervenciones en la conversación.

Dialogflow, de forma nativa, dispone de un gran número de entidades que pueden resultar muy útiles en el desarrollo de determinados agentes. Ya que este proyecto es muy específico y está centrado solo en la Comunidad de Madrid, Dialogflow no dispone de las entidades necesarias para el desarrollo. Es por este motivo por el que se han creado diferentes entidades.

Ya que el agente ha de ser capaz de reconocer los nombres de estaciones y de edificios, se han creado diferentes entidades que contienen dicha información. Otra de las entidades necesarias para el correcto funcionamiento es la denominada como "distrito". Esta entidad contiene los diferentes distritos de la comunidad de Madrid. Es necesario que el agente reconozca estos parámetros para poder pasarlos al *webhook* y que pueda realizar la consulta de manera efectiva, ya que es uno de los parámetros necesarios para ello.

distrito SAVE

☒ Define synonyms ☐ Allow automated expansion

CHAMBERI	CHAMBERI, chamberi
CENTRO	CENTRO, centro
ARGANZUELA	ARGANZUELA
BARAJAS	BARAJAS, aeropuerto
CARABANCHEL	CARABANCHEL
CHAMARTIN	CHAMARTIN
CIUDAD LINEAL	CIUDAD LINEAL
FUENCARRAL-EL PARDO	FUENCARRAL-EL PARDO, FUENCA, FUENCARRAL, EL PARDO
HORTALEZA	HORTALEZA
LATINA	LATINA, LA LATINA
MONCLOA-ARAVACA	MONCLOA-ARAVACA, MONCLOA, ARAVACA
MORATALAZ	MORATALAZ
PUENTE DE VALLECAS	PUENTE DE VALLECAS
RETIRO	RETIRO
SALAMANCA	SALAMANCA
SAN BLAS-CANEJILLAS	SAN BLAS-CANEJILLAS, SAN BLAS, CANILLEJAS
TETUAN	TETUAN
USERA	USERA
VICALVARO	VICALVARO
VILLA DE VALLECAS	VILLA DE VALLECAS
VILLAVERDE	VILLAVERDE

Fig. 3.6. Asistente de creación de entidades. Entidad distrito

Para poder entender correctamente todas las entidades, y pasar correctamente el parámetro al *webhook*, se han incluido sinónimos en la mayoría de las entradas de las entidades. De esta manera, el usuario no necesita mencionar el lugar de manera textual y tal y como aparece, si no que el agente será capaz de entenderle si utiliza alguno de los



sinónimos listados.

Es posible también que el agente aumente automáticamente las entradas de las diferentes entidades según vaya interaccionando con diferentes usuarios. En el marco de este proyecto, donde los edificios, las estaciones y los distritos son los que son, no será útil ni necesario que el agente realice estas tareas. También es capaz de ir incluyendo sinónimos, pero debido a la gran cantidad de entradas que se tienen bajo edificios, más de mil, los sinónimos pueden ser creados en un edificio incorrecto e inducir a error al realizar la consulta con el servidor externo.

### 3.2.4. Integración.

De forma nativa, el propio Dialogflow tiene la posibilidad de llevar el agente a una o más de las plataformas disponibles. Estas plataformas incluyen redes sociales que cuentan con millones de usuarios tales como Facebook o Twitter. Es posible además la inclusión del agente en cualquiera de los dispositivos inteligentes de Google, desde teléfonos a altavoces o televisores.

## 3.3. API REST

*Application Programming Interface*, más conocido por su acrónimo API, es un software intermediario que permite que dos aplicaciones se comuniquen entre ellos, mediante una serie de herramientas e instrucciones[42]. Desempeña a su vez un papel importante en la seguridad de los dispositivos y de la información que existe en la red. La comunicación se realiza mediante pequeños paquetes de datos, compartiendo solo lo necesario, evitando la exposición de los datos del usuario así como la totalidad de los datos de los servidores.

En este caso, obtendremos información a través de una API REST, acrónimo de *Representational State Transfer*. Es una interfaz de aplicación de programación que utiliza protocolos HTTP, para obtener datos y realizar modificaciones en ellos.

La arquitectura de una API REST sigue una serie de reglas [43]:

- **Interfaz uniforme.** Se debe intentar crear la API de una manera lógica y jerárquica, creando una arquitectura lo más sencilla posible. Para lograr esto, se pueden utilizar ciertos recursos del protocolo HTTP, como los mensajes auto-descriptivos.
- **Peticiones sin estado.** El servidor no recogerá información acerca de la última consulta HTTP hecha por el cliente. Cada consulta es tratada de manera individual y de un modo único, no existe un historial. De esta manera se consigue un mayor rendimiento, pero para ello es necesario que la consulta del cliente contenga toda la información necesaria para que el servidor pueda interpretar la petición.
- **Cacheable.** En la web, los clientes pueden cachear las respuestas del servidor, es decir, almacenar la información sobre la respuesta para poder reutilizarla en otras

peticiones. Las respuestas se deben marcar de forma implícita o explícita como *cacheables* o no. Si la información es almacenable, mejoraremos el rendimiento en el lado del cliente.

- **Separación de cliente y servidor.** La interfaz del cliente y del servidor está separado, y su unión se produce mediante la interfaz uniforme. Mientras esta interfaz no cambie, se podrá cambiar tanto de cliente como de servidor, aumentando la escalabilidad y la interoperabilidad entre las plataformas.
- **Sistemas de capas.** El cliente puede estar conectado al servidor mediante una interfaz directa o mediante algún intermediario. Esto es irrelevante y desconocido para ambas partes. Mediante un sistema de capas además se facilita la implementación de políticas de seguridad entre diferentes capas, evitando interferencias entre diferentes niveles.
- **Codificación bajo demanda.** Mediante *applets* o el empleo de *scripts* los servidores pueden aumentar o definir funcionalidades en el cliente.

Como se ha mencionado anteriormente, el objetivo de las API es la comunicación entre aplicaciones con el fin del traspaso de datos entre ellas. Existen una serie de métodos comunes a las API, que representan el modo en el que serán completadas y las acciones que se pueden realizar. Los métodos son:

- **Método GET.** Las peticiones de este tipo son las más comunes y más usadas en la gran mayoría de páginas web. Este método es usado para recoger información de un servidor en un punto determinado de los recursos. No se produce ningún cambio en el estado de la API ni en la información que contiene. Como respuesta a este método, el cliente recibirá los objetos de la petición así como un código de respuesta indicando si ha sido efectiva o se ha producido algún error.
- **Método POST.** Se produce un cambio en la información del servidor mediante la creación de un nuevo recurso que se añadirá a los ya existentes.
- **Método PUT.** Es similar al método POST ya que se envía información al servidor, pero con la diferencia de que la acción que se lleva a cabo es la de actualizar la información de algún recurso ya existente. En caso de que no exista el recurso, se podrá crear.
- **Método DELETE.** El fin de este método es el de eliminar algún recurso de la API.

### 3.3.1. Datos Abiertos del Ayuntamiento de Madrid.

El Portal de Datos Abiertos del Ayuntamiento de Madrid está dedicado a promover el acceso a los datos del gobierno municipal e impulsar el desarrollo de herramientas creativas para atraer y servir a la ciudadanía de Madrid [44].

Desde el año 2016, se ha creado un banco de datos abiertos, accesibles a todos los ciudadanos. El objetivo de la creación de este banco de datos es permitir a los ciudadanos de la comunidad de Madrid el disponer de estos datos para utilizar estos datos y darles un valor añadido. Se pretende que el usuario individual así como las empresas para:

- Simples consultas de información.
- Darle valor añadido a la información disponible.
- Desarrollar aplicaciones y servicios.
- Crear nuevos negocios.

Existen varias maneras en las que se pueden acceder a los datos expuestos en el portal, dónde se incluyen consultas a API así como la posibilidad de descargar ficheros de datos. Además de los datos ya publicados, existe un apartado dentro del portal donde los ciudadanos pueden hacer peticiones acerca de ciertos conjuntos de datos de los que pueden querer. De esta manera, el Ayuntamiento recogerá la información de cada petición y de ésta manera los ciudadanos pueden tener a su alcance una gran cantidad de información para su propio beneficio y el beneficio general.

Otro de los objetivos de la creación del Portal de Datos Abiertos está relacionado con la necesidad de dotar de más transparencia a las Administraciones Públicas, para así ampliar la confianza que disponen las éstas instituciones por parte de la ciudadanía.

Actualmente dispone de 394 conjuntos de datos a disposición del público. Contiene información sobre temas muy variados, tales como las actividades de ocio que se van a desarrollar en la comunidad en los siguientes cien días, estadísticas sobre accidentes de tráfico en la comunidad o información sobre aparcamientos públicos.

### **3.4. Aplicación.**

Pese a las posibilidades de integración de las que dispone Dialogflow se ha tomado la decisión de incluir el agente en una aplicación que tenga más funcionalidades aparte del propio agente. Pese a que el agente sea la funcionalidad más importante de la aplicación, hay otras funcionalidades igualmente importantes y útiles. La aplicación será funcional tanto en Android como iOS permitiendo así el acceso a la aplicación a la gran mayoría de la población de la comunidad de Madrid.

#### **3.4.1. Herramientas utilizadas.**

Para el desarrollo de esta aplicación se ha utilizado un asistente web para la creación de aplicaciones. La plataforma es conocida como Thunkable X, y una de sus mayores ventajas es que las aplicaciones creadas con esta plataforma están disponibles tanto para Android como iOS.

La principal característica de esta plataforma es la forma en la que se programan las aplicaciones. No es necesario que el desarrollador conozca ninguno de los lenguajes utilizados para el desarrollo de aplicaciones en ninguno de los sistemas operativos, ya que no es necesario escribir código. En su lugar, la programación de las aplicaciones se realiza mediante bloques y componentes que se van uniendo con la lógica que decida el programador. Por tanto, el desarrollador es el encargado de organizar las pantallas, incluir los elementos necesarios y realizar las conexiones entre las pantallas y los elementos de tal manera que todos ellos sean útiles. Se encarga también de que la aplicación sea lo más visualmente agradable e intuitiva posible. La plataforma se encarga por tanto de traducir la distribución de las pantallas y la lógica de los bloques a código que los diferentes sistemas operativos puedan compilar y ejecutar.

Existen otras plataformas similares a la utilizada en este proyecto, pero una de las funcionalidades que ofrece esta plataforma y que sus competidoras no, es la facilidad de conexión con la plataforma Dialogflow. De esta manera será más fácil enviar las consultas del usuario y mostrar las respuestas del agente.

Esta plataforma permite coger proyectos de otros usuarios, y basándonos en ellos crear nuestras propias aplicaciones. En este caso, se ha decidido empezar desde cero para poder implementar las funcionalidades deseadas y organizar la aplicación de una manera sencilla e intuitiva.

## Explicación de Thunkable X

Existen dos diferentes partes de la plataforma que se deben desarrollar para el correcto funcionamiento de la aplicación. En primer lugar nos encontramos con la pestaña *Design* o Diseño. Aquí es donde el desarrollador irá incluyendo elementos mediante *drag and drop* o en español arrastra y suelta, e irá creando las diferentes pantallas de las que dispondrá la aplicación. La otra pestaña, *Blocks* o bloques, es donde el desarrollador ha de programar la lógica que moverá la aplicación.

El primer paso para crear una aplicación, es añadir los componentes necesarios desde la pantalla de diseño. Existe la posibilidad de añadir tantos componentes como queramos en cada una de las pantallas de la aplicación. Existe una gran cantidad de componentes, agrupados en: elementos para la interfaz, componentes de diseño, elementos de voz, elementos gráficos, datos, localización, sensores, elementos para compartir, identificación de usuarios y monetización de la aplicación.

Para crear la interfaz con el usuario, tenemos los siguientes elementos:

- **Botón.** Este elemento nos permite desencadenar una serie de acciones al ser pulsado. Cuando el usuario pulse sobre un botón, podremos llevarle a otra pantalla, guardar información, borrar algún elemento, etc.
- **Etiqueta.** Mediante etiquetas podemos informar al usuario acerca de las posibili-

dades que le brinda la aplicación, la explicación de algún botón, etc.

- **Introducción de texto.** Es la manera que tiene el usuario de comunicarse con la aplicación. El usuario al pulsar sobre el campo de introducción de texto verá como se despliega su teclado en la pantalla, habilitando la escritura de texto. Será una de las formas que usemos para que el usuario realice sus consultas al agente.
- **Vista de lista.** De gran utilidad si se pretende crear algún tipo de lista de información y mostrar dicha información al usuario. Las listas se pueden ir ampliando o reduciendo.
- **Visor de páginas web.** Thunkable X nos permite tener un visor de páginas web integrado en nuestra aplicación. Así, el usuario podrá acceder a la página web sin tener que abandonar nuestra aplicación.
- **Interruptor.** Para encender o apagar alguna de las características de las que dispone el programa.
- **Deslizador.** Mediante un deslizador se permite al usuario que seleccione un valor numérico de una manera sencilla.
- **Alertas.** Permite hacer saltar una alerta con el mensaje que se desee.
- **Icono de carga.** Especialmente útil para indicar al usuario que se esta llevando alguna acción, pero que ésta está llevando más tiempo del esperado.

Todos estos elementos, al ser agregados a la pantalla deseada tienen la posibilidad de ser personalizados por el desarrollador. Por ejemplo, en el caso del botón, se podrá elegir el tamaño, los colores, el rótulo, la situación dentro de la pantalla, etc. Para que los componentes que se incluyen en la pantalla tengan una utilidad funcional en la aplicación, dentro de la pantalla "bloques", al añadir uno de estos componentes, automáticamente se añaden opciones de lógica relacionadas con este componente. Esta parte se verá más adelante en la explicación de esta pantalla.

Entre los elementos de diseño y organización de las pantallas encontramos:

- **Pestaña superior de navegación.** Permite al usuario conocer todas las pantallas de las que dispone la aplicación. El título o icono de las pantallas estará situado en la zona superior de la aplicación. Todas estas pantallas están al mismo nivel, es decir, tienen la misma importancia dentro de la aplicación. Existe la posibilidad de cambiar la apariencia del título de la pestaña activa, haciendo mucho más intuitiva la navegación por la aplicación.
- **Pestaña inferior de navegación.** El concepto de esta organización de aplicación es el mismo que el anterior, con la única diferencia es que en lugar de en la parte superior, el usuario encontrará la información sobre las pantallas en la parte inferior.

- **Stack navigator o navegación en forma de pila.** La navegación en forma de pila permite al desarrollador implementar, de una manera sencilla, la navegación entre pantallas primarias y secundarias. El usuario será dirigido desde la pantalla primaria a la secundaria al pulsar en algún botón de ésta pantalla, y podrá regresar a la pantalla principal pulsando el botón de retroceso.
- **Fila.** Este componente es utilizado para poner diferentes elementos dentro de la misma fila.
- **Columna.** Al incluir este elemento en nuestra pantalla podremos situar elementos unos encima de otros.

Los elementos bajo el título de voz son los siguientes:

- **Sonido.** Permite a la aplicación reproducir sonidos cuando se activa el bloque.
- **Text-to-speech.** Permite a la aplicación transformar textos en voz, para facilitar así la comunicación con el usuario.
- **Speech recognizer.** Este componente, también conocido como *Speech-to-text*, permite transformar en texto aquello que el usuario diga.
- **Assistant.** Como se mencionaba anteriormente, Thunkable X incluye de forma nativa inclusión con Dialogflow. Al incluir este elemento en alguna pantalla de nuestra aplicación, solo tendremos que introducir las credenciales de nuestro asistente para poder interactuar con él directamente desde la aplicación.
- **Traductor.** Permite la traducción de textos entre dos idiomas seleccionados.

En el siguiente bloque de componentes, encontramos los relacionados con imágenes y gráficos:

- **Imagen.** Permite incluir una imagen en la pantalla de nuestra aplicación. Será suficiente con subir la imagen a la plataforma y después seleccionarla para ser visualizada.
- **Librería de imágenes.** Permite al usuario abrir su biblioteca de imágenes para seleccionar una y subirla a la aplicación.
- **Cámara.** Al añadir este componente damos la posibilidad al usuario de capturar imágenes directamente desde la aplicación.
- **Reconocedor de imágenes.** Servicio conectado a los servicios cognitivos de Microsoft, permite el reconocimiento de imágenes. Al compartir una imagen con este servicio, obtendremos una descripción de la imagen, un nivel de confianza para dicha descripción y unas etiquetas relacionadas con la imagen. En el caso de que no sea capaz de reconocer la imagen devolverá un error.

- **Animación.** Este componente impulsado por Lottie permite añadir fácilmente animaciones a nuestra aplicación. Éstas pueden ser creadas o descargadas de su librería en formato .json.

Estos elementos son conocidos dentro de la aplicación como elementos invisibles. Reciben esta denominación ya que no son elementos que puedan ser vistos por el usuario en la aplicación, como puede ser un botón. En su lugar, le dan a la aplicación una serie de características que pueden facilitar mucho la accesibilidad para el usuario, pero no son visibles dentro de la aplicación, si no que es labor del desarrollador hacer que sean funcionales. Son también invisibles para el desarrollador dentro de la propia plataforma, porque al incluirlas en alguna pantalla no aparecen dentro de la pantalla, si no que solo estarán presentes en la zona bajo el título de "componentes invisibles" y dentro de la lógica de la pantalla "bloques". Existen otros componentes que se incluyen también dentro de esta sección, como las alarmas, las notificaciones o los sensores.

Dentro de los componentes relacionados con datos, encontramos la posibilidad de realizar conexiones con diferentes bases de datos, locales o en la nube, entrar en el almacenamiento local del dispositivo del usuario, realizar conexiones con API. Se pueden añadir operaciones relacionadas con monedas virtuales mediante una cartera *Blockchain*. Como último elemento encontramos contratos inteligentes, desarrollado por Oasis. Este componente nos permite realizar operaciones de visualización y envío de documentos de carácter confidencial o con información sensible.

Por otro lado tenemos la posibilidad de añadir mapas en alguna de nuestras pantallas, así como la posibilidad de acceder a la localización en tiempo real del usuario. De esta manera podremos mostrarle al usuario, basándonos en su ubicación real, rutas a lugares que desee u otra información como lugares cercanos.

Disponemos de la opción de añadir un temporizador en nuestra pantalla, con un tiempo definido por el desarrollador. De esta manera podemos activar eventos basándonos en un temporizador que se activará cuando sea necesario.

Finalmente nos encontramos con las funciones sociales, que permiten compartir ciertos elementos que de nuestra aplicación, la posibilidad de crear una aplicación donde sea necesario registrarse para poder acceder, así como iniciar sesión cada vez que se abra la aplicación y elementos para poder monetizar la aplicación a través de anuncios y pagos que realiza el usuario para obtener una versión libre de anuncios.

Dentro de la pestaña de "bloques" de la plataforma es donde se crea la lógica interna de la aplicación y se integran los diferentes elementos y se relacionan entre ellos. Todos los elementos que integramos en las pantallas tienen su parte de lógica reflejada en esta pantalla, pero además de éstos elementos, tenemos otros componentes comunes a todas las pantallas y que podremos usar independientemente de los elementos que se decidan incluir. Estos elementos son:

- **Control.** Dentro de los elementos de control encontramos los elementos básicos

de programación de bucles, en los que podemos realizar iteraciones de un proceso, bloques condicionales, y los bloques necesarios para navegar a otra pantalla o para abrir una página web.

- **Lógica.** En la parte de lógica tenemos los bloques que nos permiten evaluar enunciados para saber si son verdaderos o falsos.
- **Matemáticas.** Los bloques en este apartado son los que permiten realizar operaciones matemáticas como suma, resta, multiplicación o incluso generar variables enteras entre dos números definidos por el programador.
- **Texto.** En texto tenemos todo lo relacionado con variables de texto. Podemos crear nuevas variables, buscar ciertas letras o palabras dentro de otras palabras u oraciones.
- **Listas.** Dentro de listas encontramos todo lo relacionado para crear listas de datos, inicializarlas, aumentarlas, recortarlas o eliminarlas. Los elementos de las listas pueden ser creados por el programador, o mediante una lógica más compleja se puede conseguir que sea el propio usuario el que vaya añadiendo elementos a la lista.
- **Color.** Podemos asignar colores a diferentes elementos de la pantalla en el caso de que queramos que éstos cambien en algún momento.
- **Dispositivo.** Los elementos de dispositivo nos permiten hacer vibrar el terminal del usuario cuando deseemos, obtener la hora y el minuto actuales, saber si el dispositivo tiene una conexión activa a internet o incluso saber desde que plataforma está usando el usuario nuestra aplicación.
- **Objetos.** Nos permite crear objetos desde JSON o crear JSON a partir de un objeto. También nos permite recoger el valor de alguna variable de dicho objeto.
- **Variables.** Una de las partes más importantes a la hora de programar la aplicación es la parte de las variables. En esta parte encontramos los bloques necesarios para inicializar una variable a un valor numérico o a un texto, así como cambiar su valor en cualquier momento. De igual manera, el programador puede crear tantas variables como desee. La utilidad de las variables es la de pasar información de una pantalla a otra pantalla, que de otra manera no tendríamos la posibilidad. Dentro de las variables que se pueden crear encontramos tres tipos diferentes de variables según donde se guarde su valor:
  - **Aplicación:** se guarda en la propia aplicación.
  - **Dispositivo:** se guarda el valor en el dispositivo. De esta manera se puede recoger el valor de la variable de anteriores sesiones del propio usuario. Es especialmente útil para tener valores de ajustes del usuario.



- Nube: el valor es guardado en Firebase, servicio de Google para bases de datos guardadas en la nube. De esta manera, se pueden exportar a bases de datos externos y así poder realizar tareas de recolección de información de datos y análisis.
- **Funciones.** Dentro de funciones tenemos los bloques necesarios para poder crear funciones y poder reducir el número de bloques que usaremos en el desarrollo del programa, ya que la próxima vez que sea necesaria la misma lógica, simplemente se llamará a la función.

Es importante explicar los bloques que podemos incluir al añadir el Asistente de Dialogflow a nuestra aplicación. En la pestaña de bloques, al añadirlo a la aplicación, tendremos las diferentes posibilidades de usar esta funcionalidad. Existen dos maneras en las que podemos realizar la consulta y mantener la conversación con el asistente. Como desarrolladores podemos obligar al usuario que se comunique por voz con nuestro asistente. Para ello, incluiremos el bloque que inicia la escucha, y el bloque que detiene la escucha. El bloque de iniciar la escucha devolverá un valor, la respuesta del asistente. El otro método de comunicación con el asistente es mediante texto. Para poder realizar la comunicación por texto, necesitaremos recoger el texto introducido por el usuario en una variable, que será la que enviemos al asistente en el bloque "Query". Este bloque también nos devolverá un valor, que será también la respuesta del asistente. En ambos bloques, tanto de comenzar escucha como el de realizar la consulta por texto, una vez se ha producido la comunicación, podemos activar la lógica que sigue a esta acción.

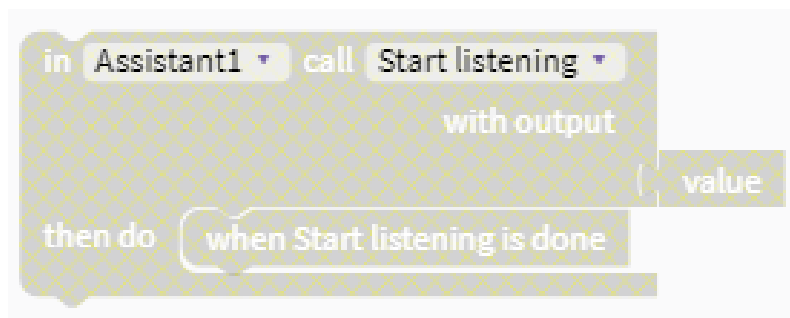


Fig. 3.7. Bloque de lógica del asistente

Mediante una combinación inteligente de todos los componentes disponibles en la plataforma, es posible crear aplicaciones intuitivas y visualmente agradables al usuario.

### 3.4.2. Presentación de la aplicación.

Antes de comenzar a desarrollar la aplicación es importante conocer cual va a ser el flujo que va a seguir un usuario al usar la aplicación.

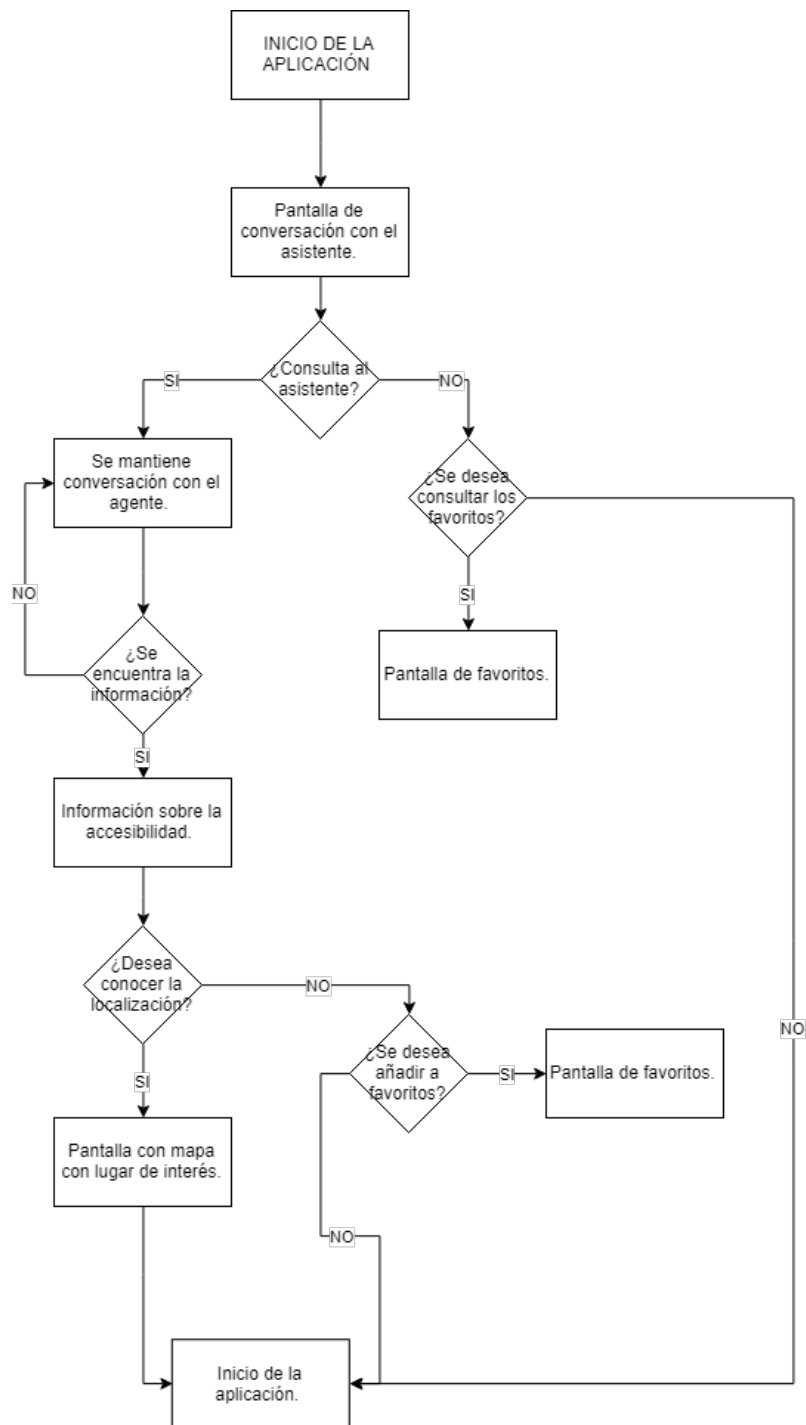


Fig. 3.8. Flujo de la aplicación

Como podemos observar en la figura 3.8, al abrir la aplicación el usuario se encuentra en la pantalla donde puede conversar con el asistente. Aquí puede decidir si desea mantener una conversación con el asistente o desea ir a la pantalla de sus lugares favoritos, donde se encuentran lugares que ha consultado anteriormente y de los que puede conocer su localización. En caso de que desee conversar con el agente, se producirán diferentes preguntas y respuestas en la conversación hasta que el asistente comprenda la consulta. Al tener los datos necesarios para realizar la consulta, devolverá la información sobre la

accesibilidad. En este momento aparecen en la pantalla dos botones diferentes que pueden ser usados por el usuario para ir a la pantalla de mapas, donde podrá conocer la ubicación del lugar en cuestión, o bien añadir este lugar en favoritos. Al añadir un lugar a favoritos se agregará a su lista personalizada de lugares favoritos, a la que podrá acceder en todo momento. En esta pantalla, podrá ver el nombre de las estaciones o edificios, y al pulsar sobre uno de ellos podrá ir al mapa para ver la ubicación, o eliminar este lugar de su lista de favoritos.

### 3.4.3. Pantalla de conversación.

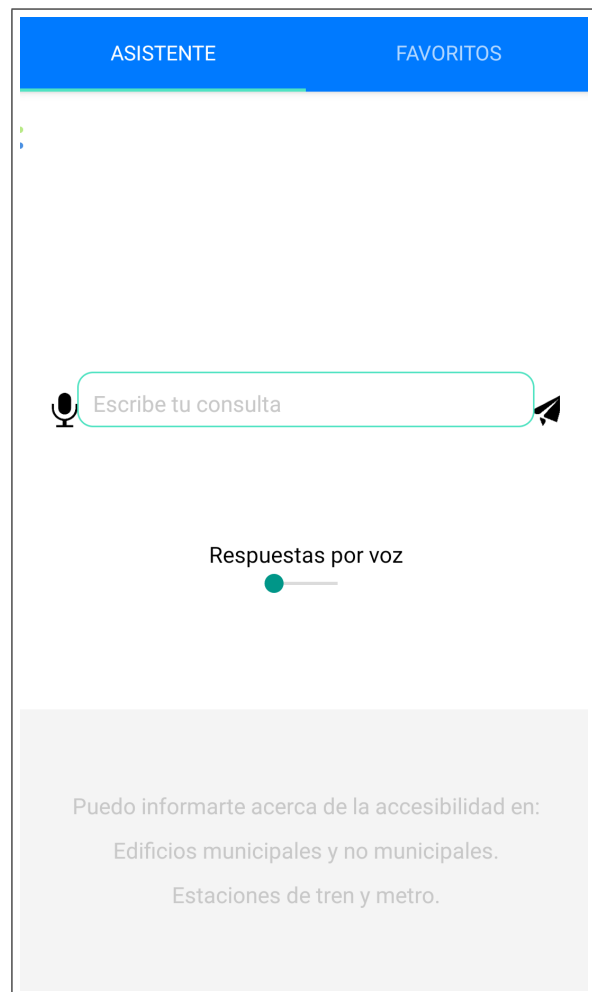


Fig. 3.9. Pantalla de inicio

En esta pantalla el usuario encuentra información escrita sobre los temas con los que puede ayudarle el asistente. Encuentra a su vez un interruptor donde puede encender o apagar las respuestas por voz por parte del asistente, en caso de que desee o no escuchar la respuesta. Encuentra un cuadro de texto donde podrá introducir las consultas al asistente, un botón para enviar el texto al asistente y un botón que activa la escucha de la aplicación y del asistente para realizar la comunicación por voz. Cuando el usuario finalice de hablar

o escribir, verá lo que ha dicho en la parte superior de la pantalla, como si se tratara de una aplicación común de mensajería. De igual manera, cuando el asistente responda, verá este texto en la parte superior, debajo de lo que él ha dicho previamente. Cada vez que el usuario o el agente realicen interacciones, se sustituirá el contenido mostrado en la pantalla por el de la última iteración.

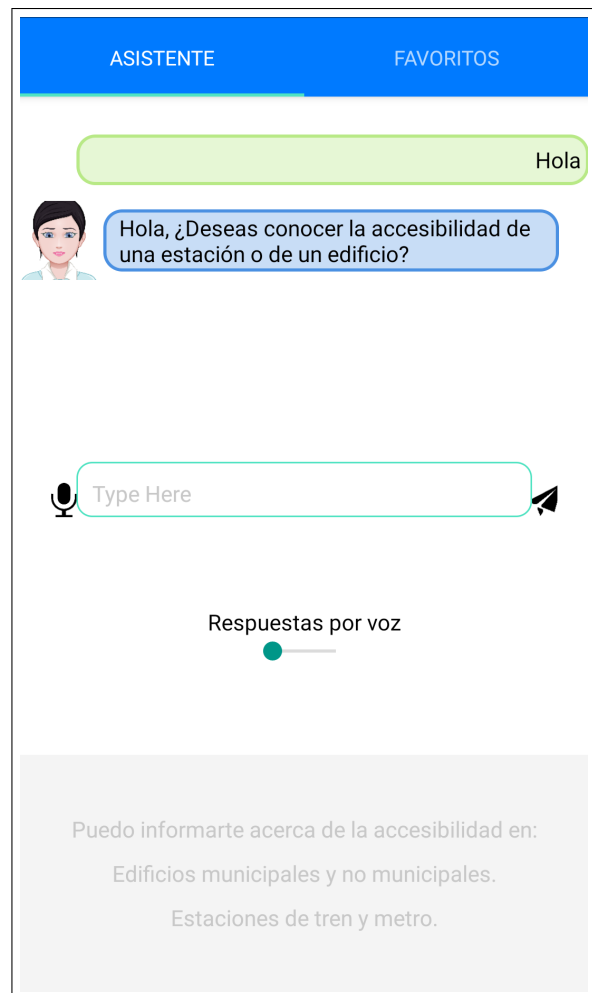


Fig. 3.10. Pantalla de conversación

Cuando termine la conversación, y el asistente haya sido capaz de realizar la consulta, y haya obtenido información sobre el edificio o la estación por la que pregunta el usuario, devolverá esta información al usuario. Cuando esto sucede, se activa una lógica dentro de la aplicación que permite mostrar una serie de botones que tienen diferentes funciones. En primer lugar, el botón de ir al mapa, le permite al usuario navegar hasta la pantalla de mapas, donde encontrará el mapa centrado sobre el lugar que acaba de ser consultado. En segundo lugar, el botón añadir a favoritos. Este botón lo que hace es añadir una nueva entrada a la lista de lugares favoritos del usuario. A parte de la lista con los nombres de los lugares, existe otra lista donde se guarda la ubicación de los lugares favoritos, para poder acceder más adelante a esta información en la siguiente pantalla. Una vez termine la consulta y haya sido efectiva, el usuario podrá volver a iniciar una conversación con el

agente sobre un lugar diferente o bien salir de la aplicación.

En la parte de los bloques de esta pantalla, es donde se crean y se inicializan los valores de todas las variables utilizadas en las diferentes pantallas de la aplicación. El modo de obtener las direcciones de los diferentes lugares, es mediante la búsqueda de esta información dentro de la respuesta proporcionada por el asistente. Este valor es guardado en una variable al pulsar el botón ir al mapa, que pasará este valor a la siguiente pantalla, permitiendo mostrar esta ubicación.

#### 3.4.4. Pantalla de lugares favoritos.



Fig. 3.11. Pantalla de favoritos

En esta pantalla se presenta en forma de lista los nombres de los diferentes lugares consultados por el usuario en anteriores conversaciones con el agente y en anteriores sesiones. Para ello se ha creado una variable lista que puede ser utilizada en diferentes pantallas. Los elementos se añaden desde la pantalla de la conversación y son visualizados desde la pantalla de favoritos. Por tanto, el valor de las variables será guardado dentro del dispositivo.

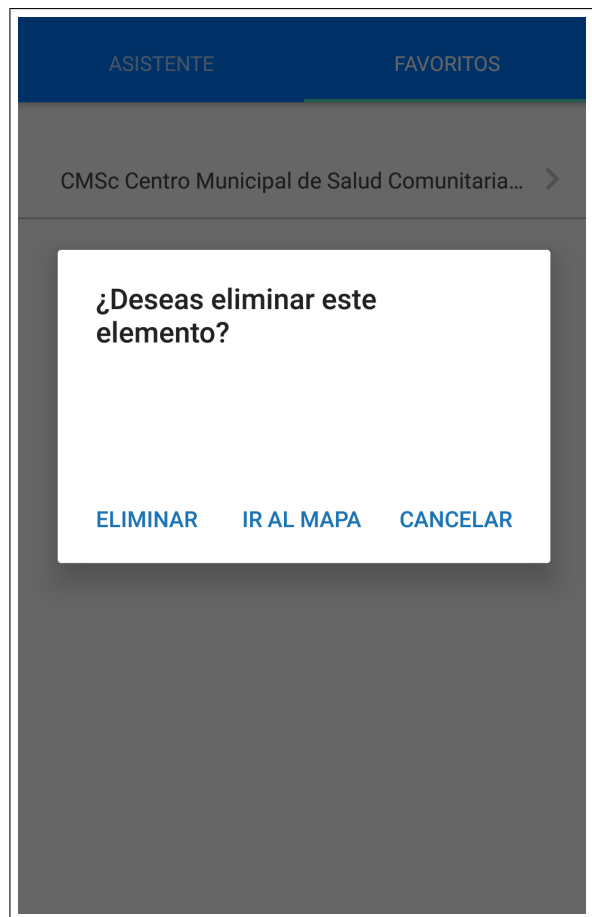


Fig. 3.12. Menú desplegable al pulsar sobre favorito

Cada elemento de la lista es a su vez un botón que activa una alerta que se le muestra al usuario. Esta alerta tiene diferentes opciones:

- **Eliminar.** Elimina la entrada de la lista que se corresponde al elemento pulsado. Este proceso es irreversible y elimina también la entrada correspondiente a la ubicación del lugar de la lista que guarda estos valores.
- **Ir al mapa.** Al pulsar este botón se accede a la lista con la información de la ubicación de los edificios y estaciones, y el usuario es llevado a la pantalla del mapa a la localización del lugar.
- **Cancelar.** En caso de que se haya producido una pulsación accidental o no decida hacer ninguna de las opciones anteriores, el usuario tiene la posibilidad de cancelar y volver a la ventana de favoritos sin realizar ninguna acción.

### 3.4.5. Pantalla de mapas.



Fig. 3.13. Pantalla de mapas

Esta pantalla muestra en un mapa la ubicación del edificio o de la estación, así como la ubicación en tiempo real del usuario. Esta pantalla es solo accesible a través de otras pantallas, y siempre que se inicie, se situará en el lugar de interés. De esta manera se evitan problemas con posibles malentendidos en el mapa debido a pulsaciones indebidas u localizaciones incorrectas. El usuario solo podrá ser dirigido a lugares por los que ha consultado al asistente y éste ha sido capaz de localizar y recabar información sobre la accesibilidad.

### 3.5. Evaluación.

En esta sección se procede a explicar las pruebas realizadas individualmente por el desarrollador y analizar los resultados obtenidos en dichas pruebas. La siguiente fase de pruebas, desarrollada más adelante en la sección, consiste en una encuesta cumplimentada por una serie de usuarios testadores de la aplicación. Estos usuarios tuvieron la posibilidad de probar la aplicación y sus diferentes funcionalidades, y más adelante, fueron

preguntados una serie de cuestiones.

### Pruebas en la aplicación.

Para comprobar el correcto funcionamiento de la aplicación antes de ser probada por otros usuarios, se han llevado a cabo una serie de pruebas con diferentes escenarios con el objetivo de valorar las diferentes funcionalidades. El funcionamiento de la aplicación previamente establecido ha de ser el que tenga cuando un usuario vaya a utilizarla. Para ello se han creado diferentes pruebas, desde las más sencillas hasta algunas más complejas pasando por todos los casos de uso.

ID de prueba	P - XX
Nombre	
Descripción	
Pasos	
Resultado esperado	

Tabla 3.1. Tabla de pruebas.

Los diferentes escenarios serán detallados mediante tablas como la tabla 3.1. Cada prueba tendrá un identificador y un nombre. Se adjuntará una breve descripción de la prueba, el resultado esperado así como los pasos necesarios para reproducir la prueba.

ID de prueba	P - 01
Nombre	Iniciar aplicación desde Android.
Descripción	Al pulsar en el icono de la aplicación, ésta debe ser capaz de abrirse.
Pasos	Clic en el icono de la aplicación desde un dispositivo Android.
Resultado esperado	La aplicación se abre con normalidad.

Tabla 3.2. P - 01: Iniciar aplicación desde iOS



<b>ID de prueba</b>	<b>P - 02</b>
<b>Nombre</b>	Iniciar el asistente por texto.
<b>Descripción</b>	Mediante entrada por texto, el asistente debe activarse.
<b>Pasos</b>	El usuario escribirá un saludo inicial que despertará al asistente.
<b>Resultado esperado</b>	Tras enviar el saludo inicial, el texto del usuario se deberá mostrar en pantalla, así como el saludo del asistente.

Tabla 3.3. P - 02: Iniciar el asistente por texto

<b>ID de prueba</b>	<b>P - 03</b>
<b>Nombre</b>	Iniciar el asistente por voz.
<b>Descripción</b>	Despertar al asistente mediante la voz.
<b>Pasos</b>	El usuario se comunicará mediante la voz con el asistente.
<b>Resultado esperado</b>	Se mostrará por pantalla la comunicación realizada por el usuario. El asistente ha de responder al saludo con otro saludo.

Tabla 3.4. P - 03: Iniciar el asistente por texto

<b>ID de prueba</b>	<b>P - 04</b>
<b>Nombre</b>	Recibir respuesta por texto.
<b>Descripción</b>	Como respuesta a cualquier comunicación, el asistente deberá responder.
<b>Pasos</b>	El usuario se comunicará por voz o por texto con el asistente.
<b>Resultado esperado</b>	La aplicación mostrará por pantalla la respuesta del asistente.

Tabla 3.5. P - 03: Respuesta del asistente por texto

<b>ID de prueba</b>	<b>P - 05</b>
<b>Nombre</b>	Recibir respuesta por voz.
<b>Descripción</b>	La respuesta del asistente será leída por la aplicación.
<b>Pasos</b>	Primero el usuario se comunicará por voz o por texto con el asistente. Tendrá que tener activado mediante el slider la comunicación por voz.
<b>Resultado esperado</b>	La aplicación mostrará por pantalla la respuesta del asistente y leerá dicha respuesta..

Tabla 3.6. P - 03: Recibir respuesta por voz

<b>ID de prueba</b>	<b>P - 07</b>
<b>Nombre</b>	Consultar información sobre edificio.
<b>Descripción</b>	El usuario desea conocer la accesibilidad en un edificio.
<b>Pasos</b>	Tras el saludo inicial, el usuario indicará que desea conocer la accesibilidad de un edificio. Después especificará cuál es el edificio deseado.
<b>Resultado esperado</b>	El usuario recibe una respuesta acerca de la accesibilidad del edificio.

Tabla 3.7. P - 07: Consultar información sobre edificio

<b>ID de prueba</b>	<b>P - 08</b>
<b>Nombre</b>	Consultar información sobre estación.
<b>Descripción</b>	El usuario desea conocer la accesibilidad en una estación.
<b>Pasos</b>	Tras iniciar la conversación, el usuario indicará que desea conocer la accesibilidad de una estación. Después especificará el nombre de la estación.
<b>Resultado esperado</b>	El usuario recibe una respuesta acerca de la accesibilidad de la estación.

Tabla 3.8. P - 08: Consultar información sobre estación

<b>ID de prueba</b>	<b>P - 09</b>
<b>Nombre</b>	Éxito en la consulta.
<b>Descripción</b>	Después de preguntar por un edificio o estación recibe la información deseada.
<b>Pasos</b>	Después de los saludos, y de preguntar por un edificio o estación, e indicar el nombre del lugar, recibe una respuesta del asistente.
<b>Resultado esperado</b>	La respuesta del asistente indica la accesibilidad de la estación o del edificio así como su localización y la posibilidad de añadir a la pestaña de favoritos.

Tabla 3.9. P - 09: Éxito en la consulta

<b>ID de prueba</b>	<b>P - 10</b>
<b>Nombre</b>	Visualizar en el mapa.
<b>Descripción</b>	Ver el lugar de la consulta en la pantalla de mapas.
<b>Pasos</b>	Después de saludarse, preguntar por un edificio o estación y especificar el nombre, se pulsa sobre el botón de ir al mapa.
<b>Resultado esperado</b>	Tras pulsar sobre el botón, el usuario será llevado hasta la pantalla de mapas, donde encontrará centrado el lugar de la consulta.

Tabla 3.10. P - 10: Visualizar en el mapa

<b>ID de prueba</b>	<b>P - 11</b>
<b>Nombre</b>	Añadir lugar a favoritos.
<b>Descripción</b>	El usuario desea añadir un lugar a favoritos tras su consulta.
<b>Pasos</b>	Después del éxito en encontrar la accesibilidad del lugar, el usuario pulsará sobre el botón de añadir a favoritos.
<b>Resultado esperado</b>	En la pantalla de favoritos el nuevo lugar añadido será el primero de la lista.

Tabla 3.11. P - 11: Añadir lugar a favoritos

<b>ID de prueba</b>	<b>P - 12</b>
<b>Nombre</b>	Ir a la ventana de favoritos.
<b>Descripción</b>	El usuario desea ir a la ventana de favoritos desde la ventana de la conversación con el asistente.
<b>Pasos</b>	En cualquier momento, el usuario puede deslizar e ir a la pantalla de la derecha o bien pulsar sobre el nombre de la pantalla.
<b>Resultado esperado</b>	Se muestra la pantalla de favoritos con los lugares de favoritos.

Tabla 3.12. P - 12: Ir a la ventana de favoritos

<b>ID de prueba</b>	<b>P - 13</b>
<b>Nombre</b>	Información sobre lugares favoritos disponible.
<b>Descripción</b>	Después de añadir lugares a favoritos y cerrar la aplicación, si ésta se vuelve a abrir, se deben guardar los lugares.
<b>Pasos</b>	Guardar un lugar en favoritos y cerrar la aplicación. Volver a abrir la aplicación e ir a la pantalla de favoritos.
<b>Resultado esperado</b>	En la ventana de favoritos se muestran todos los lugares guardados en sesiones anteriores.

Tabla 3.13. P - 13: Información sobre lugares favoritos disponible

<b>ID de prueba</b>	<b>P - 14</b>
<b>Nombre</b>	Eliminar lugar de la lista de favoritos.
<b>Descripción</b>	El usuario ya no está interesado en un lugar que tenía guardado en favoritos y desea eliminarlo de la lista.
<b>Pasos</b>	Una vez guardado el lugar, ir a la ventana de favoritos. Al pulsar sobre el lugar en cuestión, saldrá una alerta. Pulsar sobre el botón de eliminar.
<b>Resultado esperado</b>	La estación o edificio se eliminan de la pantalla de favoritos.

Tabla 3.14. P - 14: Eliminar lugar de la lista de favoritos

<b>ID de prueba</b>	<b>P - 15</b>
<b>Nombre</b>	Ir al mapa desde favoritos.
<b>Descripción</b>	Desde la ventana de favoritos, visualizar la localización de uno de los lugares guardados en favoritos.
<b>Pasos</b>	. Tras pulsar sobre uno de los lugares, saldrá una alerta. Pulsar sobre ir al mapa.
<b>Resultado esperado</b>	La aplicación se desplaza a la pantalla de mapas donde se mostrará la accesibilidad del edificio o estación en cuestión.

Tabla 3.15. P - 15: Ir al mapa desde favoritos

<b>ID de prueba</b>	<b>P - 16</b>
<b>Nombre</b>	Cancelar pulsación en pantalla de favoritos.
<b>Descripción</b>	Al pulsar sobre una estación o edificio en la ventana de favoritos, tener la posibilidad de no hacer nada.
<b>Pasos</b>	Pulsar sobre un lugar, y en el desplegable pulsar cancelar.
<b>Resultado esperado</b>	La aplicación no realiza ninguna operación más que volver a la pantalla de favoritos sin cambios.

Tabla 3.16. P - 16: Cancelar pulsación en pantalla de favoritos

<b>ID de prueba</b>	<b>P - 17</b>
<b>Nombre</b>	Desplazarse por el mapa.
<b>Descripción</b>	El usuario puede desplazarse por el mapa así como ampliarlo o reducirlo.
<b>Pasos</b>	En la pantalla de mapas, moverse por el mapa, ampliar y reducir el mapa.
<b>Resultado esperado</b>	El mapa se mueve, se amplía o se reduce sin generar incidencias.

Tabla 3.17. P - 17: Desplazarse por el mapa

<b>ID de prueba</b>	<b>P - 18</b>
<b>Nombre</b>	Volver a la pantalla de conversación desde el mapa.
<b>Descripción</b>	El usuario desea salir de la pantalla de mapas para volver a la conversación.
<b>Pasos</b>	Desde la pantalla de mapas pulsar en el botón de atrás.
<b>Resultado esperado</b>	La aplicación lleva al usuario a la pantalla de la conversación.

Tabla 3.18. P - 18: Volver a la pantalla de conversación desde el mapa

### 3.5.1. Análisis de los resultados

Una vez definidas claramente las pruebas a las que se iba a someter la aplicación, podemos ver en la tabla 3.19

<b>Identificador de prueba</b>	<b>Resultado</b>
P - 01	Éxito
P - 02	Éxito
P - 03	Éxito
P - 04	Éxito
P - 05	Éxito
P - 06	Éxito
P - 07	Éxito
P - 08	Éxito
P - 09	Éxito
P - 10	Éxito
P - 11	Éxito
P - 12	Éxito
P - 13	Éxito
P - 14	Éxito
P - 15	Éxito
P - 16	Éxito
P - 17	Éxito
P - 18	Éxito
P - 19	Éxito

Tabla 3.19. Resultado de las pruebas

Se puede observar como todas las pruebas han dado un resultado exitoso. Consideramos entonces que la aplicación está preparada para pasar a la fase de prueba en usuarios potenciales

## UAT

Esta fase de pruebas es conocida como UAT o *User Acceptance Testing*. Es la última fase del proceso de teste de software y consiste en la realización de una serie de pruebas en potenciales usuarios de la aplicación para ver si la aplicación. De esta manera se pueden realizar pruebas en escenarios reales en un entorno controlado para controlar la calidad de las respuestas de la propia aplicación y de las diferentes funcionalidades que contiene a los escenarios propuestos.

Fabricante	Modelo	Sistema operativo	Version SO
Apple	iPhone 6s	iOS	12.2
Huawei	P20 Pro	Android	9
Huawei	P10	Android	8
Apple	iPhone 7	iOS	12.2
Xiaomi	Mi8	Android	9
Apple	iPhone X	iOS	12.2.4
Apple	iPhone 6	iOS	11.2.6
Xiaomi	MiA1	Android	9
Xiaomi	MiA2 Lite	Android	9
Huawei	P20 Lite	Android	8
Apple	iPhone 7	iOS	12
Samsung	S8	Android	9
BQ	Aquaris XPro	Android	8.1.0
Xiaomi	Redmi 4X	Android	8.0

Tabla 3.20. Terminales usados en UAT

Para la realización de las pruebas se han utilizado diferentes terminales pertenecientes a cada uno de los usuarios que han realizado la aplicación. Como la aplicación funcionará tanto en Android como en iOS, se han realizado pruebas en ambos sistemas operativos, con diferentes versiones de los mismos para así cubrir un mayor espectro de terminales y tener una mayor confianza a la hora de lanzar la aplicación.

Más allá de las pruebas realizadas, todos los usuarios que han probado la aplicación han rellenado una encuesta con sus opiniones personales acerca del funcionamiento de la aplicación así como una serie de preguntas para evaluar el grado de aceptación que al aplicación tendría entre los usuarios potenciales para comprobar si sería utilizada.

El conjunto de Figuras 3.21 - 3.31 muestran los resultados obtenidos en cada pregunta.

**Pregunta 1:** ¿Le gustaría una aplicación para conocer la accesibilidad de edificios y estaciones?

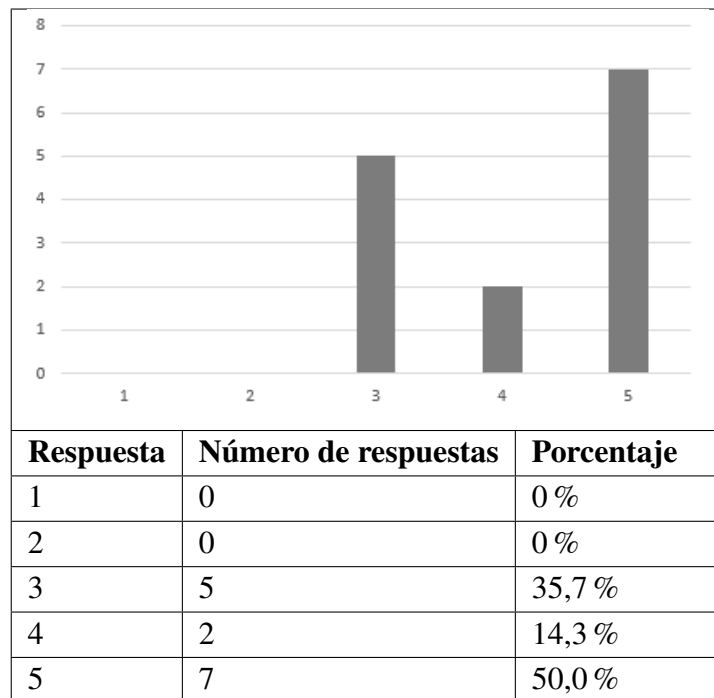


Tabla 3.21. Pregunta 1 encuesta de evaluación

**Pregunta 2:** ¿Alguna vez ha sufrido algún problema relacionado con la accesibilidad (por ejemplo: escaleras mecánicas averiadas, ascensor no funcional..)?

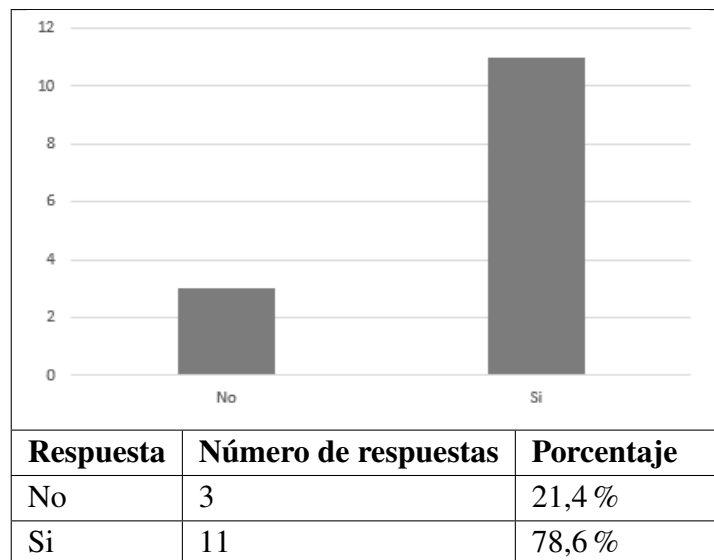


Tabla 3.22. Pregunta 2 encuesta de evaluación



**Pregunta 3:** ¿Le parece interesante la posibilidad de interaccionar por voz o por texto con el asistente?

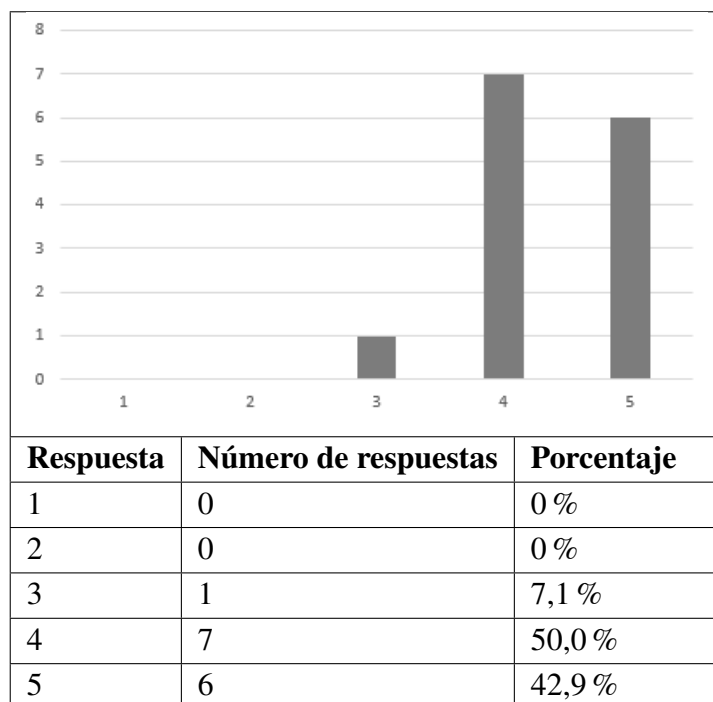


Tabla 3.23. Pregunta 3 encuesta de evaluación

**Pregunta 4:** ¿Le parece interesante poder recibir información tanto por escrito como por voz?

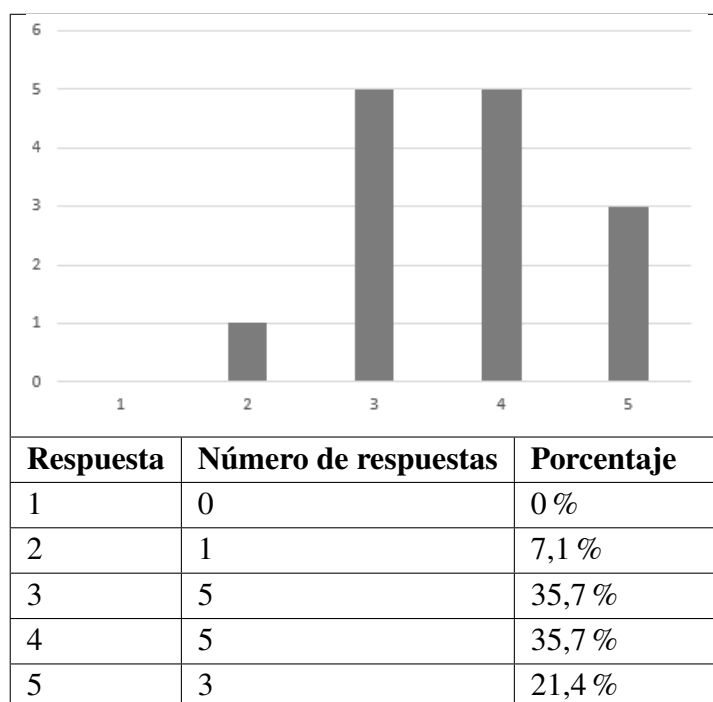


Tabla 3.24. Pregunta 4 encuesta de evaluación

**Pregunta 5:** ¿El asistente ha sido capaz de proporcionar información sobre su consulta?

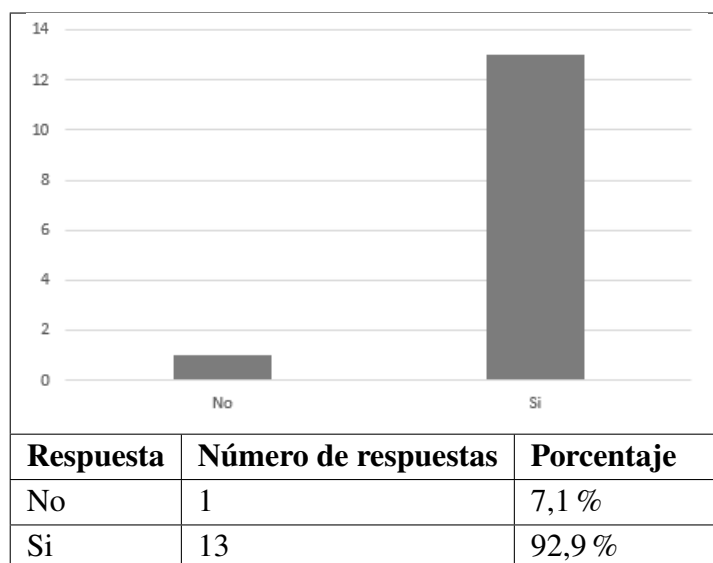


Tabla 3.25. Pregunta 5 encuesta de evaluación

**Pregunta 6:** ¿Le ha parecido útil la opción de guardar lugares destacados para tenerlos más a mano?

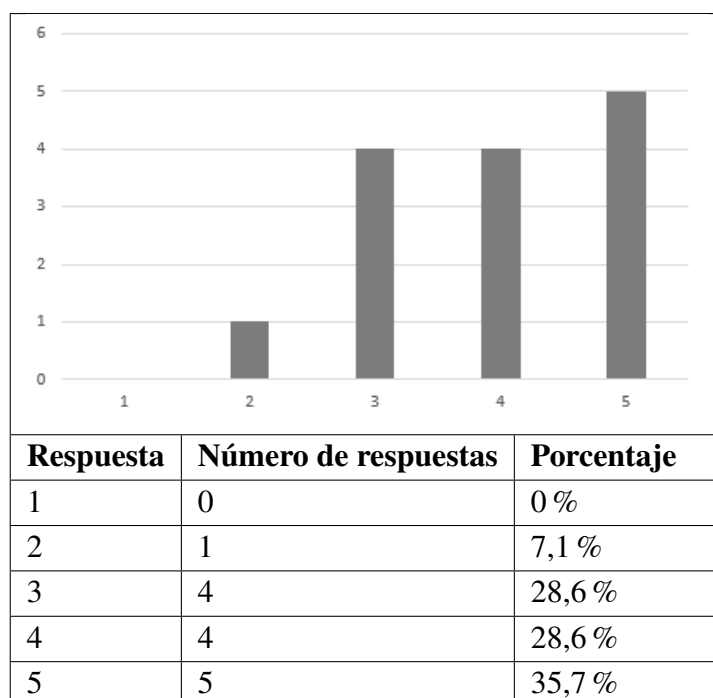


Tabla 3.26. Pregunta 6 encuesta de evaluación

**Pregunta 7:** ¿Le ha parecido interesante la opción de ver la localización del lugar de la consulta?

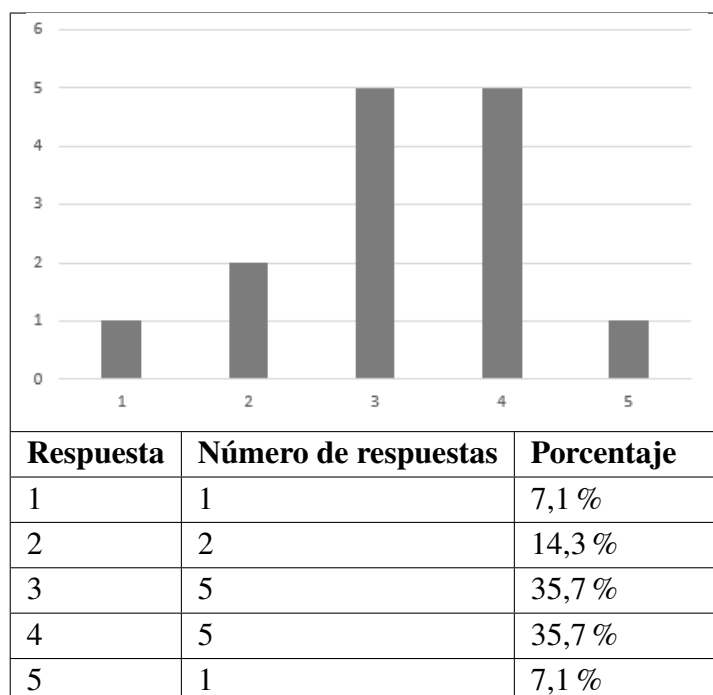


Tabla 3.27. Pregunta 7 encuesta de evaluación

**Pregunta 8:** ¿Le ha resultado fácil la interacción con el asistente?

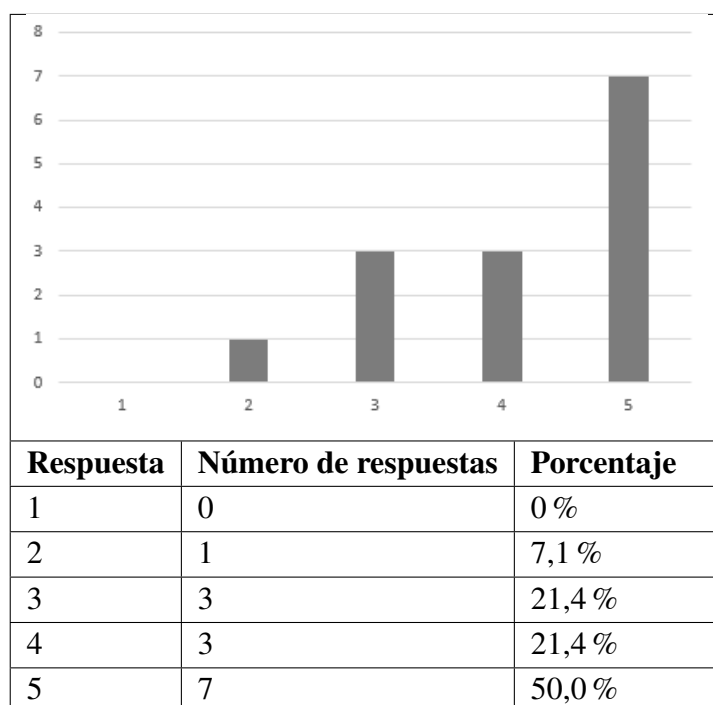


Tabla 3.28. Pregunta 8 encuesta de evaluación

**Pregunta 9:** ¿Qué nota global le pondría a la aplicación?

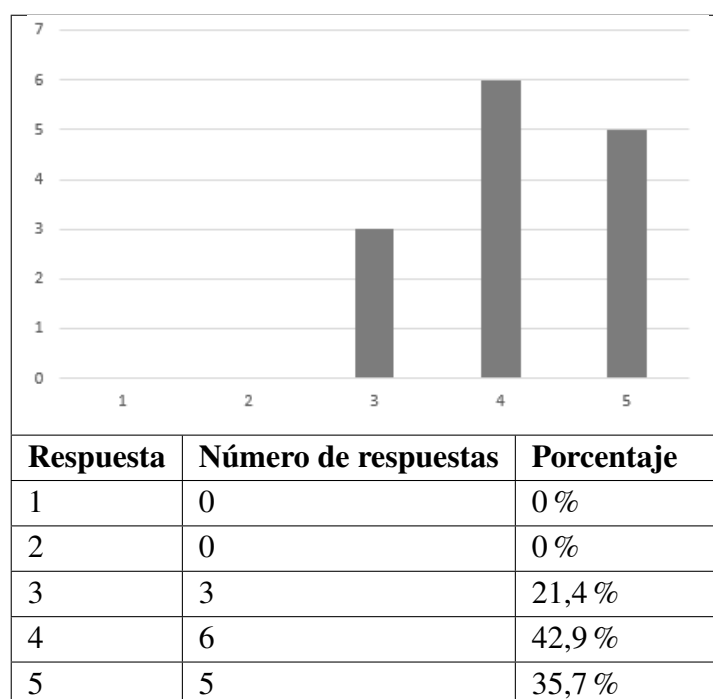


Tabla 3.29. Pregunta 9 encuesta de evaluación

**Pregunta 10:** ¿Cree que la aplicación puede ayudar a mejorar la accesibilidad en la Comunidad de Madrid?

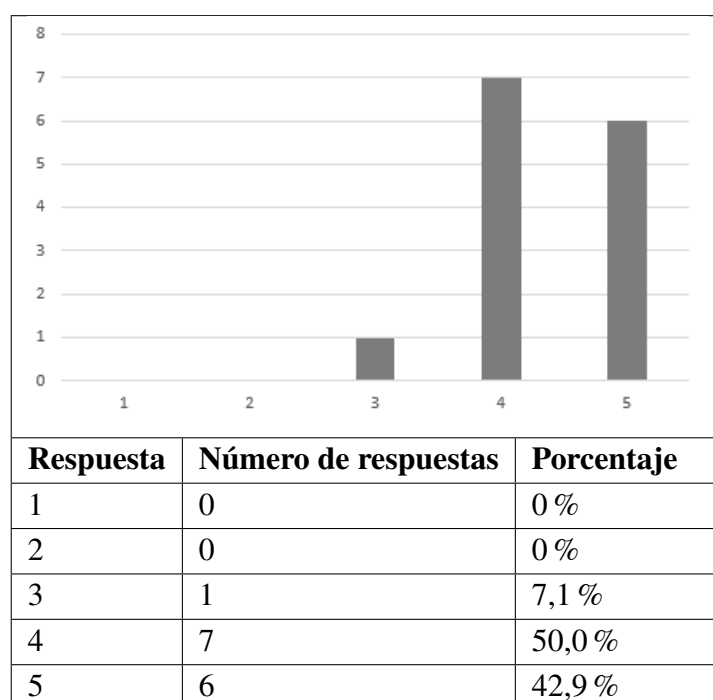


Tabla 3.30. Pregunta 10 encuesta de evaluación

**Pregunta 11:** ¿Qué probabilidades hay de que recomiende la aplicación?

Probabilidad	Número de respuestas
1	0
2	0
3	0
4	0
5	0
6	1
7	2
8	3
9	6
10	2

Tabla 3.31. Pregunta 11 encuesta de evaluación

### 3.5.2. Análisis de la UAT

Las pruebas y la encuesta ha sido realizada por un total de catorce personas diferentes, cada una con su propio terminal. Todas estas personas han sido informadas del objetivo de la aplicación. Posteriormente han procedido a la descarga de la aplicación y la instalación. Finalmente han hecho un recorrido completo por la aplicación, probando las diferentes funcionalidades. Una vez el usuario ha terminado de probar la aplicación, el cuestionario presentado en la sección anterior fue presentado.

Como observamos en la primera pregunta, una gran mayoría de los participantes tiene una opinión favorable a la creación de una aplicación con el objetivo de mejorar la accesibilidad, ya que en torno al 80 % de los encuestados se ha encontrado al menos una vez con algún inconveniente en las instalaciones en estaciones o edificios.

En cuanto a la accesibilidad de la aplicación, los participantes en la evaluación encuentran más interesante el poder hablar y escribir a tener la posibilidad de que el asistente sea capaz de leer la respuesta aparte de verla por pantalla.

El asistente por su parte obtiene una puntuación positiva en cuánto a la capacidad de resolución de las consultas de los usuarios. Un 93 % de los usuarios han obtenido información acerca de su consulta, y un 71,4 % de los usuarios han calificado de fácil la interacción con el asistente.

Por su parte, las funcionalidades de la aplicación, pantalla de favoritos y mapas han recibido resultados positivos. La mayoría de los usuarios han resultado útil la funcionalidad de lugares recientes. Cabe destacar que en torno a un 21 % de los usuarios no considera que la posibilidad de visualizar el lugar de la petición en un mapa de utilidad.

La nota global que recibe la aplicación es de un 4,14 sobre 5, una calificación muy

alta,

Como medidor del nivel de satisfacción se ha recurrido al análisis de la pregunta "¿Qué probabilidades hay de que recomiende la aplicación?" mediante NPS o *Net Promoter Score* [45]. El objetivo de éste índice es el de medir la lealtad del usuario con la aplicación desarrollada en este Trabajo de Fin de Grado. Para llevar a cabo este análisis, se presenta al encuestado una escala del 1 al 10, siendo 1 muy improbable y 10 altamente probable.

Se procede a dividir a las personas en tres grupos diferentes:

- **Promotores.** Las personas que han dado un puntaje de 9 o 10 son conocidos como promotores, y son aquellos usuarios que recomendarían la aplicación.
- **Pasivos.** Los usuarios pasivos han respondido con un 7 o un 8, y son aquellos que están satisfechos con la aplicación pero no maravillados.
- **Detractores.** Son aquellos usuarios que no están satisfechos con la aplicación y podrían realizar comentarios negativos acerca de la aplicación causando un impacto negativo.

Para calcular el NPS de la aplicación se aplica la siguiente fórmula:

$$\text{NPS} = \% \text{ de promotores} - \% \text{ de detractores} \quad (3.1)$$

Un muy buen resultado sería obtener un NPS superior al 50 %. Un NPS entre 30 - 50 % es un buen resultado, pero indica que hay ciertos usuarios que no están muy satisfechos. Obtener un NPS entre el 0 y el 30 % sería aceptable y significaría que aún hay una gran oportunidad de mejora. Finalmente, obtener un NPS negativo significa que la mayoría de los usuarios están descontentos con el producto [46].

Se obtiene un NPS de:

$$\text{NPS} = 57 \% - 7 \% = 50 \% \quad (3.2)$$

Podemos concluir entonces que la satisfacción en general es elevada y que una gran cantidad de usuarios recomendarían la aplicación a amigos o compañeros.

## 4. GESTIÓN DEL PROYECTO

En este apartado se procede a definir la gestión y la planificación del proyecto. Se explicará la gestión del tiempo y las diferentes etapas por las que ha pasado el proyecto. Se especificará el presupuesto que hubiera sido necesario para llevar a cabo este proyecto; también se explicará la situación del proyecto dentro de unos marcos regulador y socio-económico.

### 4.1. Planificación temporal

Para poder detallar la planificación temporal con la que se comenzó el proyecto, es necesario explicar en primer lugar las diferentes fases por las que el proyecto debía pasar para poder llegar a su finalización dentro del tiempo estimado. Las etapas del proyecto son las siguientes:

- **Fase 1. Documentación y análisis.**

En esta primera fase del proyecto, se llevó a cabo la investigación de todos los módulos que componen el proyecto. En primer lugar, se descompuso el proyecto en diferentes partes, analizando qué conocimientos sobre qué elementos eran necesarios para poder llevar a cabo el proyecto. Fue necesaria una documentación de las diferentes opciones a la hora de desarrollar tanto el asistente conversacional. Se tuvieron en cuenta tanto las funcionalidades que ofrecía cada plataforma, así como los costes asociados a cada una de las opciones. Una vez se decidieron las plataformas se procedió a realizar un estudio exhaustivo donde se crearon diversos agentes y otras aplicaciones de menor complejidad, donde se fueron adquiriendo los conocimientos necesarios para afrontar el proyecto real.

- **Fase 2. Desarrollo del sistema.**

Una vez terminada la documentación y la adquisición de los conocimientos necesarios para afrontar el proyecto, se pasó al desarrollo de los diferentes sistemas que lo componen. Se comenzó desarrollando el agente conversacional, para después llevar a cabo los *webhooks* necesarios para que el asistente fuera funcional. Finalmente se desarrolló la aplicación donde iría incluido el agente así como una ampliación de las funcionalidades que el propio agente incluiría de ir por separado sin aplicación.

- **Fase 3. Pruebas y evaluación.**

Una vez terminado el desarrollo se llevaron a cabo una serie de pruebas para evaluar la efectividad de los diferentes sistemas. Las pruebas fueron realizadas en primer lugar por el desarrollador, evaluando tanto el agente como la aplicación. Una vez todos los posibles fallos fueron corregidos, se pasó a realizar pruebas con potenciales usuarios para testear la aplicación en un entorno real controlado.

#### ■ Fase 4. Documentación y redacción de la memoria.

El último paso es la redacción de éste documento. Para la correcta explicación de todos los apartados, así como para poder enmarcar este proyecto dentro de las ciudades inteligentes, compararlo con otros proyectos existentes y evaluarlo ha sido necesario la consulta de nueva documentación diferente a la que se necesitó al comenzar el proyecto.

En primer lugar se ha realizado una estimación del tiempo empleado, en días, para cada una de las fases del proyecto. Se toma como inicio el día siguiente al día de la adjudicación del proyecto y se estima como fecha final del proyecto el día de la entrega de este documento.

Etapas del proyecto	Fecha de inicio	Fecha fin
Etapas 1. Documentación y análisis	20/10/2018	13/01/2019
Etapas 2. Desarrollo del sistema	10/01/2019	29/04/2019
-Etapas 2.1. Desarrollo de Dialogflow	10/01/2019	10/04/2019
-Etapas 2.2. Desarrollo de la aplicación	20/03/2019	29/04/2019
Etapas 3. Pruebas y evaluación	10/05/2019	30/05/2019
Etapas 4. Documentación y redacción	15/04/2019	14/06/2019

Tabla 4.1. Fechas de inicio y fin de las etapas del proyecto

Para poder mostrar de una manera sencilla la planificación temporal del proyecto, conociendo el comienzo y la duración de cada una de las fases se ha utilizado un diagrama de Gantt.

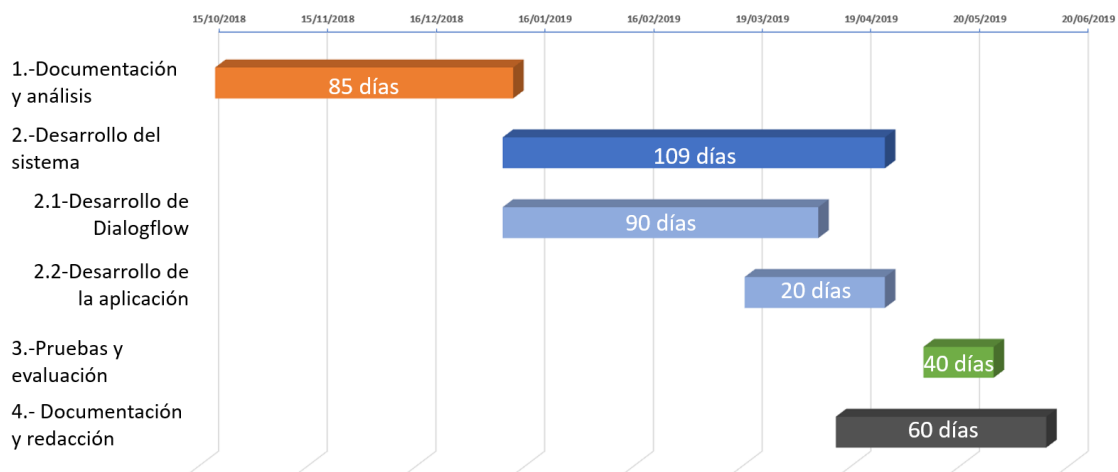


Fig. 4.1. Diagrama de Gantt

Pese a la planificación realista que se realizó sobre la carga de trabajo y su distribución en el tiempo, en determinados momentos se han producido ciertos problemas que han provocado retrasos. Éstos problemas a su vez han sido motivo de que en algunos momentos se estuvieran realizando diferentes tareas de forma simultánea.



Otro de los motivos por los que se ha alargado el tiempo previsto para la realización del proyecto ha sido la ampliación de las funcionalidades establecidas inicialmente.

Pese a la ampliación del tiempo estimado para el desarrollo del trabajo, se ha conseguido terminar el trabajo dentro de la fecha máxima de entrega del mismo.

En la tabla 4.2 podemos observar una estimación del tiempo empleado aproximadamente por día en cada una de las etapas por las que ha pasado el proyecto.

<b>Etapas del proyecto</b>	<b>Días empleados</b>	<b>Horas por día</b>
Etapas 1. Documentación y análisis	85	2
Etapas 2. Desarrollo del sistema	109	2
Etapas 3. Pruebas y evaluación	20	1.5
Etapas 4. Documentación y redacción	60	3
<b>Total</b>	<b>237</b>	<b>598</b>

Tabla 4.2. Estimación de horas empleadas

## 4.2. Marco Regulatorio

En esta sección se define el marco legislativo en el que se encuentra este proyecto. Principalmente se valorará el estado del proyecto en cuanto a la legislación vigente ,en el momento del desarrollo y redacción de la memoria, acerca de la privacidad de los usuarios y la seguridad del sistema.

En el momento existen dos regulaciones acerca de la protección de datos, una a nivel estatal y otra a nivel europeo. A nivel estatal tenemos la Ley Orgánica de Protección de Datos (LOPD) [47], mientras que a nivel europeo tenemos El Reglamento General de Protección de Datos (RGPD en español, más conocido por sus siglas en inglés GDPR) [48]. Ambos entraron en vigor en el año 2018, y amplían el espacio de seguridad del consumidor en lo relacionado con la información personal de los usuarios de diferentes sistemas o aplicaciones su seguridad [49], [50].

En primer lugar, debemos distinguir dos tipos de usuarios dentro de este proyecto. En primer lugar nos encontramos con el desarrollador del agente y de la aplicación, que es usuario de diferentes plataformas necesarias para la consecución del proyecto. Por otro lado, tenemos el usuario final de la aplicación, que será el ciudadano que decida descargar la aplicación desde uno de las tiendas virtuales de aplicaciones para poder usarla desde su propio terminal.

Para poder desarrollar el asistente conversacional es necesario aceptar los términos y condiciones de uso de Dialogflow. Éstos términos y condiciones de uso son los mismos que los que utiliza Google. Permiten monitorizar la información recibida en sus APIs para su futuro empleo en mejorar sus servicios. Éstos términos son también iguales para el servicio *Speech-To-Text* que se incluye dentro del agente y dentro de la aplicación. La

otra plataforma que ha sido empleada ha sido Thunkable X, utilizada para el desarrollo de la aplicación. Para poder utilizar esta plataforma es necesario aceptar una serie de términos de uso así como una normativa de privacidad propia. Como situación curiosa dentro de esta plataforma, en caso de que se decida usar la versión gratuita de la misma, los proyectos creados por el propio desarrollador serán de carácter público. Esta característica permite al resto de usuarios de Thunkable X el acceder a nuestros proyectos creados para visualizarlos, probarlos o incluso usarlos como base para sus propios proyectos. En el caso de que se desee tener proyectos privados, dónde se controla quién tiene la capacidad de visualizarlos, de probarlos y se evita que los demás usuarios puedan usar nuestro proyecto como base para otros, es necesario pagar por el uso de la plataforma una determinada cuota mensual.

Más adelante para poder publicar la aplicación desarrollada en las tiendas virtuales del sistema operativo Android y del sistema iOS, es necesario crear una cuenta en cada uno de los portales. Para ello tendremos que dar el consentimiento a determinados términos y condiciones de uso. Finalmente, para poder publicar la aplicación ésta deberá respetar las condiciones de privacidad establecidas así como unos términos propios de seguridad, privacidad y de propiedad intelectual de la aplicación. Por tanto, será necesario aceptar las políticas de Google y de Apple para la distribución de la aplicación en sus respectivas tiendas de aplicaciones.

Para que el usuario pueda disfrutar de la aplicación en su terminal será necesario que disponga de una cuenta en la tienda de aplicaciones desde donde se pretende descargar la aplicación. El desarrollador no ha incluido ningún servicio de recogida y recopilación de datos de los usuarios de la aplicación, siendo el único dato que se queda guardado el de los lugares favoritos, datos que pueden ser eliminados por el propio usuario y serán eliminados al desinstalar la aplicación. Por este motivo, el usuario no necesita aceptar ningún término específico en cuanto a la protección de sus datos dentro de la aplicación. El usuario no tendrá que aceptar los términos y las condiciones relativas al asistente conversacional ya que ya han sido aceptados por el desarrollador en el momento de la creación de la cuenta en Dialogflow. No existe ningún otro dato que se pregunte al usuario de carácter personal y que pueda vulnerar ninguno de los términos expuestos en la LOPD ni en el GDPR, por tanto, la aplicación cumpliría con ambas normativas.

### **4.3. Presupuesto**

El cómputo del presupuesto se ha basado en la planificación temporal realizada en apartados anteriores. De esta forma se ha podido calcular de una forma bastante cercana a la realidad lo que costaría llevar a cabo un proyecto de este tipo.

Para el cálculo de los costes se ha distinguido entre dos tipos diferentes de costes: los costes directos y los costes indirectos.

#### 4.3.1. Costes directos

Dentro de los costes directos podemos encontrar el cálculo del presupuesto necesario para poder pagar al desarrollador de la aplicación. Para el cálculo del sueldo del empleado se ha tenido en cuenta que se trata de un programador recién graduado con poca o ninguna experiencia laboral previa, por lo que, según estiman páginas de empleo como Indeed, éste sueldo se sitúa en torno a unos 20 €/hora

Personal	Horas estimadas	Sueldo por hora	Sueldo total
Alejandro Pérez Dueñas	598 horas	15 €/ hora	8970 €

Tabla 4.3. Estimación del sueldo del empleado

Dentro de los costes directos se incluyen a su vez los costes de los recursos que han sido necesarios para el desarrollo del proyecto, tanto costes de hardware como de software.

Elemento	Precio con IVA
Surface Pro 3 (Ordenador)	850 €
Teclado Surface Pro 3	150 €
Monitor hp	179 €
<b>Total</b>	1179 €

Tabla 4.4. Estimación del coste del Hardware

Ninguno de los componentes ha sido empleado únicamente para el desarrollo del proyecto, por lo que la vida útil de los diferentes componentes es mayor a la duración total del proyecto. Para poder poner en términos reales de costes asociados al proyecto, es necesario calcular la amortización de todos los componentes. Ya que tanto el ordenador, el teclado y el monitor son componentes con una elevada calidad, suponemos una vida útil de éstos de cuatro años. Para poder calcular la amortización utilizaremos la siguiente fórmula:

$$\text{amortización} = \frac{t}{T} \text{coste} \quad (4.1)$$

Donde t representa el tiempo en meses de uso de los componentes, T la vida útil en meses y Coste el precio pagado por los componentes. Por tanto tenemos:

$$\text{amortización} = \frac{10}{48} 1179 = 245,63€ \quad (4.2)$$

Obtenemos una amortización de 245,63 € para los componentes usados en el desarrollo de la aplicación. Hemos de tener en cuenta los costes asociados a su vez al dispositivo móvil utilizado para la realización de las pruebas, un Motorola Moto G5 Plus. De la misma forma que sucede con los demás componentes, la vida útil del terminal es mayor a la duración del proyecto, pero en este caso la vida útil se sitúa en dos años. Teniendo en cuenta

que se incurrió en un desembolso de 279 € en el momento de la compra, calculamos la amortización de la siguiente manera:

$$\text{amortización} = \frac{10}{24} 279 = 116,25€ \quad (4.3)$$

Por último, encontramos como costes de software, los costes asociados al empleo de las diferentes aplicaciones y plataformas para el desarrollo tanto del agente como de la aplicación. En este caso, ya que se han utilizado las versiones gratuitas en estos programas, tan sólo se tendría que tener en cuenta el gasto asociado a Microsoft Excel.

Software	Precio
Dialogflow	0 €
Thunkable X	0 €
Microsoft Office 365	99 €
<b>Total</b>	<b>99 €</b>

Tabla 4.5. Estimación del precio del software

#### 4.3.2. Costes indirectos

Los costes indirectos representan aquellos costes que no son imputables directamente como costes del proyecto, si no que se refieren a costes asociados para el correcto desarrollo del proyecto, como puede ser el coste de la electricidad, del transporte físico del trabajador, etc. Como estimación, los costes indirectos suponen un 20 % de los costes directos. Los costes indirectos ascienden entonces a:

$$\text{Costes indirectos} = 0,2 * \text{costes directos} = 0,2 * 9430,88 = 1886,18€ \quad (4.4)$$

#### 4.3.3. Costes totales

Los costes totales suponen la suma de los costes directos más los costes indirectos:

Tipo de coste	Coste
Capital humano	8970 €
Equipamiento de Hardware	361,88 €
Equipamiento de Software	99 €
Costes indirectos	1886,18 €
<b>Total</b>	<b>11317,06 €</b>

Tabla 4.6. Estimación de los costes totales

#### **4.4. Entorno socio-económico**

El objetivo de este apartado es el de realizar un análisis acerca de las posibilidades económicas y de expansión de la aplicación desarrollada en este proyecto. Para ello se analizarán datos acerca del interés que puede suscitar la aplicación entre los potenciales usuarios.

En primer lugar, es necesario realizar un análisis acerca de cuáles y quienes son los potenciales usuarios de ésta aplicación. En esta primera fase del desarrollo, la aplicación solo es funcional a pequeña escala, es decir, en el marco de la comunidad de Madrid, pero no hay que perder de vista donde se enmarca el proyecto. El proyecto se sitúa como un elemento más de una ciudad inteligente, enmarcado en un proyecto común europeo que tiene como objetivo mejorar la calidad de vida en las ciudades apoyándose en las ICT. Pese a que el desarrollo actual en esta materia sea lento y con pocos avances realmente visibles, existen una gran cantidad de prototipos para ciudades, y el número de ciudadanos de las ciudades no para de crecer, por lo que el número de ciudadanos de las futuras ciudades inteligentes y conectadas será de grandes dimensiones. Por tanto, con el modelo actual de aplicación y con las posibilidades que brinda, los usuarios potenciales que darían uso a esta aplicación serán los ciudadanos de las ciudades conectadas con algún tipo de dificultad para la movilidad y que necesitan de ciertos elementos externos para poder moverse libremente por las ciudades, tanto por edificios como para usar el transporte público. Debemos tener en mente que un número elevado de usuarios del sistema de transporte público son visitantes de la ciudad, que carecen de conocimientos acerca del propio transporte así como de la accesibilidad de las diferentes estaciones.

Por tanto, el impacto que los servicios de la aplicación y el número de usuarios que pueden beneficiarse de ellos, por el momento, es pequeño. Pese a esto, éste numero puede crecer enormemente con la expansión de la aplicación a otras urbes españolas y europeas.

A nivel español existen 3,85 millones de personas con algún tipo de discapacidad, y en la figura 4.2 podemos observar como se distribuye en la pirámide de población.

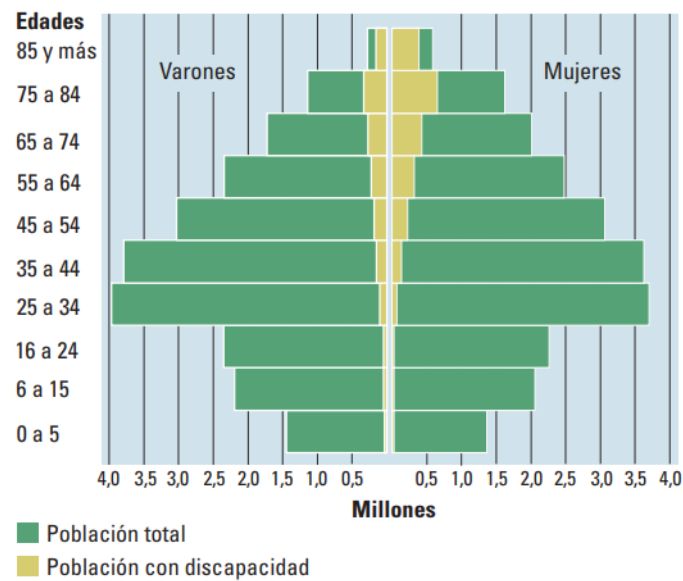


Fig. 4.2. Población con discapacidad en España [51]

Estos datos nos indican que la mayor parte de las personas con diversidad funcional aumenta a medida que se aumenta la edad de la población. El hecho de que la aplicación sea accesible en si misma y permita la comunicación a través de texto y voz, permitirá a los usuarios de avanzada edad el utilizar la aplicación en su beneficio. Es posible también que las personas encargadas del cuidado de las personas de avanzada edad hagan uso de la aplicación para planificar salidas con estas personas.

Por otro lado, los potenciales usuarios de esta aplicación no son solo las personas con dificultad a la hora de moverse, si no que personas con problemas temporales o incluso personas que lleven a niños en carritos pueden beneficiarse de los servicios que trae consigo la aplicación.

## 5. CONCLUSIONES Y TRABAJO FUTURO

EL desarrollo de este Trabajo de Fin de Grado ha permitido ampliar los conocimientos que tenía acerca del desarrollo y de las diferentes formas de hacer que este no se produzca de una forma caótica, si no de una forma controlada y organizada, teniendo en cuenta diferentes aspectos, desde el desarrollo económico, la movilidad de las personas, las comunicaciones o incluso el control de la contaminación.

Existen dos bloques en los que me gustaría dividir el proyecto, para así poder analizar cada uno de ellos por separado. Los diferentes bloques son: investigar acerca del concepto de ciudades inteligentes y las repercusiones que los proyectos que se enmarcan en este concepto traen consigo; y la investigación y el desarrollo de diferentes sistemas operativos, formas de crear aplicaciones y el desarrollo de agentes conversacionales.

El número de noticias relacionadas con el aumento de la población, el aumento de la contaminación a nivel global, tanto en emisiones de contaminantes gaseosos como de otros contaminantes como puede ser el plástico, el envejecimiento de la población es inmenso. Diariamente la población es informada mediante periódicos, noticias en la televisión o programas de radio acerca de todos estos problemas y más. Es por esto que se estén llevando a cabo medidas en las grandes ciudades para atacar estos problemas. Todos estas soluciones se enmarcan dentro de un concepto muy amplio y que engloba muchos aspectos, el de las ciudades inteligentes.

Antes de realizar este trabajo, mi concepción de las ciudades inteligentes difería ciertamente a la concepción que tengo ahora, el momento de finalización del proyecto y después de meses de investigación. El término ciudad inteligente, nos recuerda a otros conceptos que tenemos muy arraigados en nuestra sociedad, como televisión inteligente o teléfono inteligente. Es quizás por esto, que el concepto que parte de la sociedad tiene sobre las ciudades inteligentes esté directa y solamente relacionado con la conectividad de los ciudadanos entre ellos. No obstante, el término de ciudad inteligente engloba a todos los avances, proyectos, políticas y medidas que pretendan mejorar la calidad de vida de las personas en las urbes a través de las tecnologías de la información y la comunicación.

Ya que el concepto de ciudad inteligente es tan amplio, se pretende paliar los problemas crecientes en medio ambiente, sostenibilidad, movilidad e integración de las personas a través de las diferentes tecnologías de las que se dispone en la actualidad. Por tanto, no sólo se pretende crear un mundo más conectado, si no que se pretende aprovechar los avances en tecnología, comunicación e información para cambiar el modo de vida de la sociedad, redirigirlo hacia un modo de vida más sostenible y eficiente, donde los problemas que causan tantos inconvenientes a día de hoy queden a un lado.

Pese a que gran parte de las medidas que se han de tomar deben venir por parte de los gobiernos, es papel de todos los ciudadanos el tomar parte en estas iniciativas o crear unas

propias para ayudar en esta materia. Este Trabajo de Fin de Grado pretende marcar un avance en la información de la que disponen los ciudadanos de la Comunidad de Madrid, presentando información al alcance de todos de una manera más accesible y en un formato más útil y agradable para todos.

El segundo bloque consiste en la investigación y el desarrollo de las dos partes que se compone el trabajo, el agente conversacional y la aplicación en la que está integrado. La relevancia que están tomando los agentes conversacionales en los últimos años en la sociedad es de un nivel muy elevado. Se está convirtiendo en un elemento básico en una gran cantidad de compañías de diferentes sectores, permitiendo a estas compañías prestar un mejor servicio a sus clientes a menor precio de coste.

El hecho de que sean capaces de aprender según interaccionan con los usuarios, añadido al desarrollo del *machine learning*, del *big data* y del *deep learning*, van a permitir que los agentes conversacionales sean integrados en otros aspectos de la sociedad y de la vida diaria de las personas. Se están desarrollando los primeros robots conversacionales para grandes hipermercados; en hospitales serían de gran utilidad para manejar las urgencias, guiar a personas por el hospital o incluso ayudar en tareas de rehabilitación en determinados pacientes.

El poder conocer de primera mano como se desarrollan los agentes conversacionales, analizando las posibles preguntas de los usuarios y las respuestas que los usuarios esperan a estas preguntas ha cambiado, en parte, la forma en la que veo a los asistentes que existen en la actualidad. La complejidad de desarrollo de los agentes conversacionales transversales, es decir, los asistentes que no tienen un solo objetivo y son capaces de realizar diferentes tareas y contestar preguntas de diversos ámbitos, es muy elevada. Hay que tener en cuenta los casos de uso que se le pueden dar a los agentes, y cuánto mas casos tengamos, más complejo será la programación del asistente.

También, el desarrollar una aplicación en la que integrar el asistente ha sido útil para pensar en la lógica que siguen muchas aplicaciones que millones de usuarios tienen descargados en sus teléfonos móviles. La simpleza, la intuitividad y la funcionalidad de la aplicación son tres conceptos que han de estar presentes en el desarrollo de una aplicación. No es difícil encontrarse con aplicaciones repletas de funcionalidades ocultas, o a las cuáles no se puede acceder de un modo intuitivo, causando una sensación negativa en el usuario final.

## **5.1. Trabajo futuro**

En este proyecto se ha desarrollado una aplicación que cuenta con un agente conversacional capaz de proporcionar información acerca de la accesibilidad de las estaciones y de los edificios municipales y no municipales en la Comunidad de Madrid. Ya que es un proyecto que se podría incluir dentro del Horizonte 2020, existen posibles escenarios de ampliación en un futuro que producirían un gran impacto.



La ampliación del proyecto a otras ciudades Españolas o Europeas ofrecería más información a un mayor número de usuarios potenciales de la aplicación. Dentro de la misma aplicación, se podría tener información relevante acerca de las ciudades de interés de cada uno de los usuarios, para poder acceder a ella de una manera sencilla. Traducir la aplicación y permitir el interaccionar con el asistente en diferentes idiomas permitiría que muchos usuarios que desconocen la lengua de una ciudad puedan acceder a información relevante sin necesidad de contar con un traductor externo.

Es posible también ampliar el proyecto en horizontal, es decir, ampliar los servicios que trae la aplicación a los usuarios. Posibles implementaciones que no han sido posibles en el momento de desarrollar este proyecto, principalmente por falta de información por parte de los organismos públicos sería el añadir la posibilidad de que los usuarios pudieran por ejemplo conocer diferentes actividades que se van a desarrollar en su ciudad y que son accesibles para ellos, desde cine con subtítulos, obras de teatro con intérpretes para sordos, etc.

## BIBLIOGRAFÍA

- [1] *Urban population growth*, mar. de 2015. [En línea]. Disponible en: [https://www.who.int/gho/urban\\_health/situation\\_trends/urban\\_population\\_growth\\_text/en/](https://www.who.int/gho/urban_health/situation_trends/urban_population_growth_text/en/).
- [2] *Esperanza de vida al nacer, total (años)*. [En línea]. Disponible en: <https://datos.bancomundial.org/indicador/SP.DYN.LE00.IN>.
- [3] V. Albino, U. Berardi y R. M. Dangelico, “Smart Cities: Definitions, Dimensions, Performance, and Initiatives”, *Journal of Urban Technology*, vol. 22, n.º 1, pp. 3-21, 2015. DOI: 10.1080/10630732.2014.942092. eprint: <https://doi.org/10.1080/10630732.2014.942092>. [En línea]. Disponible en: <https://doi.org/10.1080/10630732.2014.942092>.
- [4] L.-F. Hurtado, J. Planells, E. Segarra y E. Sanchis, *Spoken dialog systems based on online generated stochastic finite-state transducers*. 2016.
- [5] J. Llisterri, *Introducción a los sistemas de diálogo*.
- [6] M. Techlabs, *Can Chatbots Help Reduce Customer Service Costs by 30 %*, abr. de 2017. [En línea]. Disponible en: <https://chatbotsmagazine.com/how-with-the-help-of-chatbots-customer-service-costs-could-be-reduced-up-to-30-b9266a369945>.
- [7] *Chatbots, a Game Changer for Banking & Healthcare, Saving 8 billion dollars Annually by 2022*. [En línea]. Disponible en: <https://www.juniperresearch.com/press/press-releases/chatbots-a-game-changer-for-banking-healthcare>.
- [8] B. Kinsella, *Amazon Increases Global Smart Speaker Sales Share in Q4 2018, While Google's Rise Narrows the Gap and Apple Declines*, feb. de 2019. [En línea]. Disponible en: <https://voicebot.ai/2019/02/20/amazon-increases-global-smart-speaker-sales-share-in-q4-2018-while-googles-rise-narrows-the-gap-and-apple-declines/>.
- [9] M. Juste, *Google Home comienza a pisar los talones a Amazon Echo en la carrera de los altavoces inteligentes*, sep. de 2018. [En línea]. Disponible en: <http://www.expansion.com/economia-digital/companias/2018/09/01/5b80343d268e3e7d188b45bc.html>.
- [10] B. Kinsella, *information topics requested smart speakers May 2018 FI*, ago. de 2018. [En línea]. Disponible en: <https://voicebot.ai/2018/08/01/what-people-ask-their-smart-speakers/information-topics-requested-smart-speakers-may-2018-fi/>.

- [11] *Amazon Echo (2.<sup>a</sup> generación) - Altavoz inteligente con Alexa, tela de color gris claro.* [En línea]. Disponible en: [https://www.amazon.es/dp/B079PFKDZC?tag=hydes-21&ref=pd\\_sl\\_devechoes](https://www.amazon.es/dp/B079PFKDZC?tag=hydes-21&ref=pd_sl_devechoes).
- [12] *What Is Alexa? What Is Amazon Echo, and Should You Get One?* [En línea]. Disponible en: <https://thewirecutter.com/reviews/what-is-alexa-what-is-the-amazon-echo-and-should-you-get-one/>.
- [13] K. Murnane, *What Is Deep Learning And How Is It Useful?*, abr. de 2016. [En línea]. Disponible en: <https://www.forbes.com/sites/kevinmurnane/2016/04/01/what-is-deep-learning-and-how-is-it-useful/#3e774aa0d547>.
- [14] *Google Home.* [En línea]. Disponible en: [https://store.google.com/es/product/google\\_home](https://store.google.com/es/product/google_home).
- [15] T. Bell y T. Bell, *3 Ways Cities Use Chatbots*, oct. de 2018. [En línea]. Disponible en: <https://www.cmswire.com/digital-workplace/3-ways-cities-use-chatbots/>.
- [16] . [En línea]. Disponible en: <https://consulta.renfe.com/irene/main;jsessionid=EB204CA3B175F4136201C7A0C4968DF3>.
- [17] *Dialogflow vs Rasa - Which One to Choose?*, ago. de 2018. [En línea]. Disponible en: <https://hackernoon.com/dialogflow-vs-rasa-which-one-to-choose-844c42117cb2>.
- [18] Nobbot, *Desarrollar chatbots puede resultar más fácil usando Dialogflow*, mayo de 2019. [En línea]. Disponible en: <https://www.nobbot.com/general/desarrollar-chatbots-usando-dialogflow/>.
- [19] *Dialogflow Pricing.* [En línea]. Disponible en: <https://dialogflow.com/pricing>.
- [20] *Fulfillment overview | Dialogflow.* [En línea]. Disponible en: <https://dialogflow.com/docs/fulfillment>.
- [21] *A brief introduction to Chatbots with Dialogflow.* [En línea]. Disponible en: <https://www.margo-group.com/en/news/a-brief-introduction-to-chatbots-with-dialogflow/>.
- [22] *Intents overview | Dialogflow.* [En línea]. Disponible en: <https://dialogflow.com/docs/intents>.
- [23] *Entities overview | Dialogflow.* [En línea]. Disponible en: <https://dialogflow.com/docs/entities>.

- [24] K. Kourtit, P. Nijkamp y D. Arribas, “Smart cities in perspective – a comparative European study by means of self-organizing maps”, *Innovation: The European Journal of Social Science Research*, vol. 25, n.º 2, pp. 229-246, 2012. doi: 10.1080/13511610.2012.660330. eprint: <https://doi.org/10.1080/13511610.2012.660330>. [En línea]. Disponible en: <https://doi.org/10.1080/13511610.2012.660330>.
- [25] Y.-T. Chen, “SKETCH INDUSTRY PROMOTION FRAMEWORK FOR SMART LIVING SERVICES BY LEVERAGING LIVING LAB HARMONIZATION CUBE”, jun. de 2019.
- [26] *Smart cities*. [En línea]. Disponible en: <https://www.creatingsmartcities.es/smartcity.php>.
- [27] E. Tranos y D. Gertner, “Smart networked cities?”, *Innovation: The European Journal of Social Science Research*, vol. 25, n.º 2, pp. 175-190, 2012. doi: 10.1080/13511610.2012.660327. eprint: <https://doi.org/10.1080/13511610.2012.660327>. [En línea]. Disponible en: <https://doi.org/10.1080/13511610.2012.660327>.
- [28] *Mapping Smart Cities in the EU*. 2014.
- [29] F. P. Prada, G. V. Romero, V. F. Añez y J. D. Sánchez, *Movilidad Inteligente*. [En línea]. Disponible en: <https://www.mincotur.gob.es/Publicaciones/Publicacionesperiodicas/EconomiaIndustrial/RevistaEconomiaIndustrial/395/FIAMMA%20PEREZ%20y%20OTROS.pdf>.
- [30] *The smart city model*, 2015. [En línea]. Disponible en: <http://www.smart-cities.eu/?cid=2&ver=4>.
- [31] *Estrategia Europa 2020*, ago. de 2018. [En línea]. Disponible en: [https://ec.europa.eu/info/business-economy-euro/economic-and-fiscal-policy-coordination/eu-economic-governance-monitoring-prevention-correction/european-semester/framework/europe-2020-strategy\\_es](https://ec.europa.eu/info/business-economy-euro/economic-and-fiscal-policy-coordination/eu-economic-governance-monitoring-prevention-correction/european-semester/framework/europe-2020-strategy_es).
- [32] *World Urbanization Prospects The 2014 Revision*. 2014.
- [33] *Challenge and objectives*. [En línea]. Disponible en: <http://simon-project.eu/challenge-and-objectives/>.
- [34] *Portal web del Ayuntamiento de Madrid*. [En línea]. Disponible en: <https://www.madrid.es/portales/munimadrid/es/Inicio/Movilidad-y-transportes/Informacion-Proyecto-SIMON?vgnextfmt=default%5C&vgnextoid=98451081854b8510VgnVCM10000001d4a900aRCRD%5C&vgnnextchannel=220e31d3b28fe410VgnVCM10000000b205a0aRCRD>.
- [35] *2.2. Arquitectura Android - Software de Comunicaciones*. [En línea]. Disponible en: <https://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android>.

- [36] *Platform Architecture | Android Developers*. [En línea]. Disponible en: [https://developer.android.com/guide/platform#top\\_of\\_page](https://developer.android.com/guide/platform#top_of_page).
- [37] D. Calvo, *¿Qué es el kernel y para qué sirve?*, mayo de 2019. [En línea]. Disponible en: <https://www.androidpit.es/que-es-kernel-para-que-sirve>.
- [38] Carltics, *iOS y su arquitectura interna en 4 capas*, jun. de 2016. [En línea]. Disponible en: <http://blog.ticsandroll.es/ios-y-su-arquitectura-interna-en-4-capas/>.
- [39] *Arquitectura iOS - Tecnología iOS*. [En línea]. Disponible en: <https://sites.google.com/site/tecnologiaiostm/desarrollo-de-aplicaciones/arquitectura-ios>.
- [40] *Arquitectura iOS*. [En línea]. Disponible en: <http://ios-sistema.blogspot.com/p/arquitectura-ios.html>.
- [41] Intellipaat, *iOS Architecture - iOS Tutorial*. [En línea]. Disponible en: <https://intellipaat.com/tutorial/ios-tutorial/ios-architecture/>.
- [42] *REST API*. [En línea]. Disponible en: <https://restfulapi.net/rest-architectural-constraints/#cacheable>.
- [43] Juanda, *Arquitectura de una API REST*. [En línea]. Disponible en: <https://juanda.gitbooks.io/webapps/content/api/arquitectura-api-rest.html>.
- [44] *Portal de datos abiertos del Ayuntamiento de Madrid*. [En línea]. Disponible en: <https://datos.madrid.es/portal/site/egob>.
- [45] *Mejores prácticas de NPS: ¿Qué es Net Promoter Score y cómo me puede ayudar?* [En línea]. Disponible en: <https://support.zendesk.com/hc/es/articles/203759076-Mejores-pr%C3%A1cticas-de-NPS-Qu%C3%A9-es-Net-Promoter-Score-y-c%C3%B3mo-me-puede-ayudar->.
- [46] *What is a good NPS score? – Delighted*. [En línea]. Disponible en: <https://blog.delighted.com/what-is-a-good-nps-score>.
- [47] *Agencia Estatal Boletín Oficial del Estado*. [En línea]. Disponible en: <https://www.boe.es/buscar/doc.php?id=DOUE-L-2016-80807>.
- [48] *RGPD y la nueva LOPD: Protección de Datos en España En 2019*, ene. de 2019. [En línea]. Disponible en: <https://es.mailjet.com/blog/news/rgpd-lopd-proteccion-de-datos/>.
- [49] Acumbamail, *GDPR en España: todo lo que debes saber sobre el RGPD*. [En línea]. Disponible en: <https://acumbamail.com/gdpr/>.
- [50] C. Herrero, *GDPR En España: Aplícala Y Evita Que Te Sancionen*, mar. de 2019. [En línea]. Disponible en: <https://nosinmiscookies.com/gdpr-espana/>.
- [51] I. N. de Estadística, “Panorámica de la discapacidad en España”, *Boletín INE*,